MAX-PLANCK-INSTITUT FÜR INFORMATIK

Rewrite Techniques
for
Transitive Relations

Leo Bachmair
Harald Ganzinger

MPI–I–93–249 November 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).

Abstract

We propose inference systems for dealing with transitive relations in the context of resolution-type theorem proving. These inference mechanisms are based on standard techniques from term rewriting and represent a refinement of chaining methods. We establish their refutational completeness and also prove their compatibility with the usual simplification techniques used in rewrite-based theorem provers. A key to the practicality of chaining techniques is the extent to which so-called variable chainings can be restricted. We demonstrate that rewrite techniques considerably restrict variable chaining, though we also show that they cannot be completely avoided for transitive relations in general. If the given relation satisfies additional properties, such as symmetry, further restrictions are possible. In particular, we discuss (partial) equivalence relations and congruence relations.

Keywords

Automated Theorem Proving, First-Order Logic, Transitive Relations, Chaining, Rewrite Techniques, Simplification

Contents

1 Introduction

Rewrite techniques, as exemplified by completion procedures, have been successfully applied to many problems in equational theorem proving. The theoretical foundation of completion is the use of rewrite rules, guided by well-founded syntactic orderings, in such a way that certain commutation properties are satisfied. Building ordered rewriting into the inference system, through the concept of rewrite proofs, the explicit generation of the congruence closure of a set of equations is usually avoided. This is one main advantage of rewrite techniques over methods based on resolution alone. From a practical point of view the simplification techniques afforded by rewriting are also important. In the context of automated theorem proving, rewrite techniques have been typically applied to equivalence relations. In this paper we investigate their application to arbitrary (e.g., non-symmetric) transitive relations and design corresponding inference systems for first-order theories with transitive relations, using resolution as the underlying inference mechanism.

Resolution, in its different variants, forms the core of many current automated reasoning systems. By and large such resolution refinements as hyper-resolution, ordered resolution, or the set-of-support strategy are quite useful in practice, but for logical theories with transitive relations, such as logics with equality or inequality relations, they are not very effective. Special techniques have been devised for such theories: chaining and variable elimination for inequalities, and paramodulation, which can be thought of as a form of subterm chaining (cf. Section 5), for equality. Chaining essentially encodes certain resolution steps with the transitivity axiom. For example, if $<$ is a transitive relation, chaining allows one to derive $u\sigma < v\sigma$ from $u < s$ and $t < v$, where σ is a most general unifier of s and t. The completeness of this inference mechanism follows immediately from the completeness of resolution with selection (Bachmair and Ganzinger 1993c).

We propose the following refinement of chaining: first compare the term $s\sigma$ (which is identical to $t\sigma$) to both $u\sigma$ and $v\sigma$ (using a given well-founded ordering on terms) and perform the chaining inference only if $s\sigma$ is maximal. This form of "ordered chaining" can be viewed as an application of standard rewrite techniques. More precisely, we argue that it corresponds to a completion process, aimed at deriving enough rewrite rules so that the given transitive relation can be described by corresponding rewrite proofs. The completeness of ordered chaining is based on a suitable commutation property of the rewrite relations involved which is achieved through completion. In formalizing this approach we use standard methods from the theory of term rewriting. However, we wish to apply this approach not only to unit clauses, as is the case with standard completion procedures, but to general clauses, and therefore have to integrate the notion of rewriting in this more general context. This is relatively straightforward for Horn clauses, but more difficult in the case of disjunctions of positive literals, due to the additional degree of non-determinism in the rewrite relations and certain subtle dependencies between them. We will discuss the two cases separately.

The practicality of chaining methods crucially depends on the extent to which a particularly prolific form of chaining, called variable chaining (where s or t is a variable), can be restricted. Ordered chaining considerably cuts down the number of possible variable chainings, as the required maximality condition for terms can only be satisfied if the variable is unshielded, i.e., occurs only as an argument of the predicate $<$, but not as an argument of a function symbol or any other predicate symbol. However, we also show that some variable chainings are needed for transitive relations in general, though further restriction are possible if additional properties such as symmetry are satisfied. For instance, in the case of equivalence relations we obtain calculi in which chaining "into" a variable is not needed.

The paper is organized as follows. After introducing some terminology, we propose, in Section 3, ordered chaining as an inference mechanism for dealing with transitive relations and prove its refutational completeness for Horn clauses. Then completeness for general clauses is established for an extended inference system and in the presence of a notion of redundancy that covers the usual simplification techniques used in rewrite-based theorem provers. A discussion of variable chaining concludes the section. We then consider symmetric transitive relations: (partial) equivalence relations in Section 4 and congruence relations in Section 5; and conclude with a summary and suggestions for further research.

2 Preliminaries

We consider first-order languages with function symbols, predicate symbols, and variables. 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 $P(t_1, \ldots, t_n)$, where P is a predicate symbol of arity n and $t_1, \ldots, t_n$ are terms. A *literal* is an expression A (a *positive* literal) or $\neg A$ (a *negative* literal), where A is an atomic formula. 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.

3 Transitive Relations

The main question we consider in this paper is the satisfiability of sets of clauses in languages with transitive relations, that is, predicates $<$ that satisfy

$$x < y, y < z \rightarrow x < z.$$

(We usually use infix notation for binary predicates.) Such clauses present difficulties for resolution-based provers. A transitivity clause can be resolved with itself to yield a new clause

$$x < y_1, y_1 < y_2, y_2 < z \rightarrow x < z$$

that can again be resolved with the original clause, to yield

$$x < y_1, y_1 < y_2, y_2 < y_3, y_3 < z \rightarrow x < z$$

and so on. Certain refinements of resolution provide better control in this regard. For instance, the self-resolution that is problematic with transitivity, is automatically excluded by resolution with selection Bachmair and Ganzinger (1993c). More precisely, if we select the two negative literals in the transitivity axiom, the only kind of inference in which it can participate are inferences of the form

$$\frac{C\ ,\ u < s \quad D\ ,\ t < v \quad x \not< y\ ,\ y \not< z\ ,\ x < z}{C\sigma\ ,\ D\sigma\ ,\ u\sigma < v\sigma}$$

where σ is a most general unifier of s and t. The key here is that it is never necessary to resolve with the non-selected (i.e., positive) literal of the transitivity axiom (for details, see Bachmair and Ganzinger (1993c)). Therefore we may encode transitivity as an inference rule:

Chaining:

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

where σ is a most general unifier of s and t.

Unfortunately, the chaining rule, which was already proposed by Slagle (1972), is not completely satisfactory, as it explicitly generates the transitive closure of the binary relation described by $>$. For example, if $a < b$ and $b < c$ and $c < d$ are given, chaining will generate the additional unit clauses $a < c$ and $b < d$ and $a < d$.

We will next outline how standard rewrite techniques can be used in this context to more effectively restrict the chaining rule. Rewrite systems provide for a more compact representation of transitive (closures of) relations and are one of the more efficient techniques for reasoning about chains of terms, such as $a < b < c < d$ in the above example. Rewrite techniques have typically been applied to equivalence and congruence relations. We also discuss non-symmetric relations, developing an approach similar in spirit to what has been called bi-rewriting by Levy and Agustí (1993).

3.1 Commutation

If R is a binary relation, we denote by R^{-1} its inverse, by R^+ its transitive closure, and by R^* its reflexive-transitive closure. (A binary relation R is called *reflexive* if xRx for all x; *irreflexive* if xRx for no x; and *non-reflexive* if it is not reflexive.) The composition of two binary relations R and S is denoted by $R \circ S$. Our objective is to represent transitive (closures of) relations by means of well-founded rewrite relations.[1] Two relations R_1 and R_2 are employed for that purpose.

We say that R_1 *commutes* with R_2 if $R_2 \circ R_1 \subseteq (R_1^+ \circ R_2^*) \cup R_2^+$.[2]

Proposition 1 *If $R_1 \cup R_2^{-1}$ is a well-founded binary relation and R_1 commutes with R_2, then the two relations $(R_1 \cup R_2)^+$ and $(R_1^+ \circ R_2^*) \cup R_2^+$ are identical.*

The proposition can be proved with standard methods from rewriting theory.[3] It provides the starting point for our investigations.

[1]The relations we talk about are abstract reduction systems in the sense of Klop (1992), but not necessarily rewrite relations in the sense of Dershowitz and Jouannaud (1990), in that they need not be closed under context application.

[2]The relation $(R_1^+ \circ R_2^*) \cup R_2^+$ is identical to $(R_1^+ \circ R_2^*) \cup (R_1^* \circ R_2^+)$.

[3]Commutation properties, with a view to their application to termination problems, have been discussed in Bachmair and Dershowitz (1986), Bellegarde and Lescanne (1987), Geser (1991).

Let $\succ$ be a (strict) ordering and $\succeq$ be its reflexive closure. We denote by $R_\succ$ the intersection of R with $\succ$; by $R_\succeq$ the intersection of R with $\succeq$; by $R_\prec$ the intersection of R with the inverse $\prec$ of the relation $\succ$; and by $R_\preceq$ the intersection of R with the inverse $\preceq$ of $\succeq$. The relation $R_\succeq \cap R_\preceq$ describes the reflexive part of R. If the ordering $\succ$ is total, then $R = R_\succeq \cup R_\preceq$. Slightly adapting Proposition 1, we obtain:

Proposition 2 *Let R be a binary relation and $\succ$ be a well-founded ordering that is total on the given domain. If the relation $R_\succeq$ commutes with $R_\preceq$, then $R^+ = (R_\succeq \cap R_\preceq) \cup (R^+_\succ \circ R^*_\prec) \cup R^+_\prec$.*

This result shows that under suitable commutation properties the non-reflexive part of a transitive relation can be described by a certain kind of rewrite proofs with $R_\succ$ and $R_\prec$. If the two relations $R_\succeq$ and $R_\preceq$ do not commute, one may use a so-called completion process to obtain commuting relations that describe the same transitive closure. Completion always succeeds for finite ground relations, provided the given syntactic ordering $\succ$ is total.[4] Equational completion procedures, such as ordered completion or conditional completion, have been described by inference rules, e.g., Bachmair, Dershowitz and Plaisted (1989), Ganzinger (1991), Dershowitz (1991). We will devise similar (refutationally complete) inference systems for transitive relations in general.

3.2 Ordered chaining

Let $<$ be a binary predicate that is assumed to denote a transitive relation. We denote by $T_<$ the set consisting of the corresponding transitivity axiom. We call models of $T_<$ *transitivity interpretations.* Literals $u < v$ and $\neg(u < v)$ (also written $u \not< v$) are called *inequalities.* Let I be a set of ground atoms, containing inequalities and possibly other atoms, and let $\succ$ be an (syntactic) ordering on terms. We define relations $\rightarrow_I$ and $\leftarrow_I$ and $\equiv_I$ as follows. For every inequality $u < v$ in I, we have $u \rightarrow_I v$ if $u \succ v$; $u \leftarrow_I v$ if $v \succ u$; and $u \equiv_I v$ if u and v are the same. We also write $u \downarrow_I v$ if $u \rightarrow^+_I \circ \leftarrow^*_I v$ or $u \leftarrow^+_I v$. In other words, $u \downarrow_I v$ if and only if there is a non-empty sequence of rewrites

$$u = u_0 \rightarrow_I \cdots \rightarrow_I u_m = v_n \leftarrow_I \cdots \leftarrow_I v_0 = v$$

where $m + n \geq 1$. We call such a sequence a *rewrite proof* (of $u < v$). A sequence $u \leftarrow_I t \rightarrow_I v$, on the other hand, is called a *peak*, and is said to *commute* if $u \downarrow_I v$. A sequence of rewrites is a rewrite proof if and only if it contains no peak as a subsequence. By the *rewrite closure* $I^\downarrow$ of I we mean the set $I \cup \{u < v : u \downarrow_I v\}$.

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 $a \rightarrow_I b$ and $b \rightarrow_I c$ and $c \rightarrow_I d$. The relation $\rightarrow_I$ evidently commutes with the (empty) relation $\leftarrow_I$. If we choose the ordering $b \succ c \succ d \succ a$, the rewrite steps are $a \leftarrow_I b$ and $b \rightarrow_I c$ and $c \rightarrow_I d$, and there is a peak, $a \leftarrow_I b \rightarrow_I c$, that does not commute. By chaining we generate $a < c$, and the corresponding rewrite step $a \leftarrow_I c$ guarantees commutation of the peak. Chaining with suitable ordering restrictions provides the basis for a completion process of which equational completion procedures are a special case in which the two rewrite relations $\leftarrow_I$ and $\rightarrow^{-1}_I$ are the same, cf., Section 4.

When we use ordered chaining, a (ground) inequality will be considered to be true if there exists a suitable rewrite proof. In the case of theories described by (general) clauses, the notion of rewriting also has to be reflected in the handling of negative literals.

[4] For a survey on orderings see Dershowitz (1987).

Take the above example and add another clause $a < d \to d < e$. By transitivity we have $a < d$, which we may resolve with the additional clause to obtain $d < e$. However, with ordered chaining, we do not derive $a < d$ directly, but only obtain enough inequalities for a suitable rewrite proof of $a < d$. We need an inference mechanism that also generates enough inequalities for a rewrite proof of $d < e$.

We combine the chaining rules for inequalities with an ordered resolution calculus, and for that purpose need to extend the ordering $\succ$ to literals. We assume that $\succ$ is well-founded, satisfies the subterm property (i.e., $A[s] \succ s$, for all expressions A and proper subexpressions s of A), and is total on ground terms and ground atoms. We extend $\succ$ to a well-founded ordering on ground literals by associating with each such literal L a complexity measure (max_L, p_L, s_L, min_L) defined as follows. Let L be a literal A or $\neg A$. The second component p_L is $\top$, if L is negative, and $\bot$ otherwise. If A is not an inequality, then max_L is A, and s_L and min_L are both $\bot$. If A is an inequality $u < v$, then max_L denotes the maximal, and min_L the minimal, term of u and v; and s_L is $\top$ if the maximal term occurs as first argument and $\bot$ otherwise. (We assume that $\top$ and $\bot$ are new elements with $\top \succ t \succ \bot$, for all t.)

For example, if $s \succ t$, then the complexity of $s < t$ is $(s, \bot, \top, t)$, whereas the complexity of $t \not< s$ is $(s, \top, \bot, t)$. The complexity of $s < s$ is $(s, \bot, \top, s)$. The resulting ordering is total on ground literals. Its extension to (finite) multisets, which we also denote by the same symbol, is well-founded and total on ground clauses.

We assume that any of these orderings are extended to non-ground expressions as follows: $E \succ E'$ 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.[5]

We have the following chaining rules for inequalities.

Ordered Chaining:

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

where σ is the most general unifier of s and t, $u\sigma < s\sigma$ is strictly maximal in $C\sigma$, $t\sigma < v\sigma$ is strictly maximal in $D\sigma$, $u\sigma \not\succeq s\sigma$, and $v\sigma \not\succeq s\sigma$.

Negative Chaining:

$$\frac{C\,,\, u \not< s \quad D\,,\, t < v}{C\sigma\,,\, D\sigma\,,\, v\sigma \not< s\sigma}$$

where σ is the most general unifier of u and t, $u\sigma \not< s\sigma$ is maximal in $C\sigma$, $t\sigma < v\sigma$ is strictly maximal in $D\sigma$, $v\sigma \not\succeq u\sigma$, $s\sigma \not\succeq u\sigma$, and $s\sigma \neq v\sigma$; and

$$\frac{C\,,\, u \not< s \quad D\,,\, t < v}{C\sigma\,,\, D\sigma\,,\, u\sigma \not< t\sigma}$$

where σ is the most general unifier of s and v, $u\sigma \not< s\sigma$ is maximal in $C\sigma$, $t\sigma < v\sigma$ is strictly maximal in $D\sigma$, $u\sigma \not\succeq s\sigma$, $t\sigma \not\succeq s\sigma$, and $u\sigma \neq t\sigma$.

The resolution rule applies to arbitrary literals, including inequalities.

[5] 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.

Ordered Resolution:

$$\frac{C \,,\, A \quad D \,,\, \neg B}{C\sigma \,,\, D\sigma}$$

where σ is the most general unifier of A and B, $A\sigma$ is strictly maximal in $C\sigma$, and $B\sigma$ is maximal in $D\sigma$.

The calculus of all these inference rules is denoted by $\mathsf{B}_<$, or simply B.

Remark. We generally assume that the premises of an inference have no common variables. If necessary, the variables in one premise are renamed. Thus, it is also possible to use different variants of a clause as premises in one inference.

3.3 Refutational completeness for Horn clauses

Clauses with at most one positive literal are called *Horn clauses*. The calculi $\mathsf{B}_<$ are refutationally complete for such clauses. For the completeness proof we adapt the model construction approach of Bachmair and Ganzinger (1990) (see also Bachmair and Ganzinger 1993c).

Given a set N of ground clauses, we define a corresponding Herbrand interpretation I using induction on $\succ$. More precisely, we define for each clause C an interpretation I_C, intended to be a model for clauses smaller than C, and sets R_C and E_C that are designed to turn I_C into a model of C as well. Formally, for every clause C in N we define R_C to be the set $\bigcup_{C \succ D} E_D$. By I_C we denote the rewrite closure $R_C \cup \{u < v : u \downarrow_{R_C} v\}$ of R_C. Furthermore, if C is a clause $C' \vee A$, where A is a positive literal and $A \succ C'$,[6] and C is false in I_C then $E_C = \{A\}$. In that case, we also say that C is *productive* and that it *produces* A. In all other cases, $E_C = \emptyset$. Finally, let R be $\bigcup_C E_C$ and let I be the rewrite closure $R \cup \{u < v : u \downarrow_R v\}$ of R.

In what follows we shall also use the notation R^C for $R_C \cup E_C$ and I^C for the rewrite closure of R^C. Hence I_C is the partial interpretation defined by clauses smaller that C, whereas I^C additionally includes the effect of C in this construction. All these interpretations are rewrite closures of suitable rule sets.

Lemma 1 *If a ground Horn clause C (which need not be in N) is true in some interpretation I_D or I^D, where $D \succeq C$, then C is also true in I and in any interpretation $I_{D'}$ and $I^{D'}$ with $D' \succ D$. Furthermore, if a clause C is productive, then it is true in I^C.*

Proof. The proof is straightforward, once one has observed that the ordering $\succ$ is designed so that whenever a negative literal $\neg A$ occurs in C and A is false in I_D or I^D with $D \in N$ and $D \succeq C$, then A remains false in I. □

This lemma will often be applied in its contrapositive form to infer that C is false in I_C and I^C whenever it is false in $I_{D'}$ or $I^{D'}$, for some $D' \succ C$.

We say that a set of (possibly non-ground) clauses N is *saturated* (with respect to an inference system J) if it contains all conclusions of inferences (in J) from N. Let us also use $\rightarrow_I^{\equiv}$ as an abbreviation for $\equiv_I \cup \rightarrow_I$ (and similarly for $\leftarrow_I^{\equiv}$ and $\downarrow_I^{\equiv}$), for any set I of ground atoms.

[6]In writing $A \succ C'$ we use A to denote a positive unit clause.

Lemma 2 *Let N be a set of ground Horn clauses that is saturated with respect to* B *and does not contain the empty clause, and let I be the interpretation constructed from N. Then for every clause C in N, we have:*

(1) If C is a productive clause $C' \vee A$, with $A \succ C'$, then C' is false in I.

(2) The relation $\to_{R^C}^{\equiv}$ commutes with $\leftarrow_{R^C}^{\equiv}$, and hence the relation $\downarrow_{R^C}^{\equiv}$ is transitive, and I^C as well as I_C are transitivity interpretations.

(3) C is true in I^C.

Proof. We use induction on $\succ$. Let C be a ground clause in N, such that (1)-(3) are satisfied for all smaller clauses in N.

(1) Suppose C is a productive clause of the form $C' \vee A$, with $A \succ C'$. Since C is a Horn clause, C' contains only negative literals. Furthermore, C' is false in I_C. Thus, if $\neg B \in C'$, then $B \in I_C \subseteq I$, from which we conclude that C' is false in I.

(2) By the induction hypothesis, peaks that involve only R_C commute, hence I_C is a transitivity interpretation. The assertion for I^C is thus trivially true if E_C does not contain an inequality, or if $E_C = \{t < v\}$ where $v \succeq t$. Suppose $C = C' \vee v < t$ produces $v < t$ with $v \succ t$. If there is another clause $D = D' \vee u < v$ that produces $u < v$ with $v \succ u$, then there exists a peak $u \leftarrow_{R_C} v \to_{E_C} t$. But by ordered chaining we obtain from C and D the clause $C'' = C' \vee D' \vee u < t$. Since N is saturated and $C \succ C''$ we may apply the induction hypothesis, to infer that C'' is true in $I^{C''}$. But $C' \vee D'$ is false in I and $R^{C''} \subseteq R_C$, so that either $u \downarrow_{R_C} t$ or else $u = t$ and $u < t \in R_C$, which establishes the required commutation property.

(3) If C is false in I^C, then it cannot be productive, hence must violate the condition imposed on productive clauses, that is, the maximal literal in C is negative. Let us therefore assume C can be written as $C' \vee \neg A$, with a maximal atom $\neg A$ that is false in I_C. Thus the atom A is true in I_C. If A is produced by some (smaller) clause $D = D' \vee A$, then the resolvent $C'' = D' \vee C'$ of C and D is smaller than C (since $\neg A \succ D'$), contained in N, and false in I_C and hence false in $I^{C''}$—which contradicts the induction hypothesis. The only remaining possibility is that A is an inequality $u < v$ with $u \downarrow_{R_C} v$. In that case there exists a suitable productive clause $D \vee u < v'$ (where $v' \downarrow_{R_C} v$) or $D \vee u' < v$ (where $u \downarrow_{R_C} u'$), which through negative chaining with C produces a clause C'' smaller than C, that is contained in N and is false in $I^{C''}$, which is again a contradiction. Thus, C must be true in I^C, which completes the proof. □

The lemma indicates that all interpretation I_C and I are transitivity interpretations. As an immediate corollary of the lemma we obtain:

Theorem 1 *If a set of Horn clauses N is saturated with respect to $\mathsf{B}_<$, then the set $N \cup T_<$ is unsatisfiable if and only if it contains the empty clause.*

Proof. If N contains the empty clause, $N \cup T_<$ is unsatisfiable. Suppose N does not contain the empty clause. Let I be the Herbrand interpretation constructed from the set of all ground instances of N. It follows from standard results in resolution theory (usually called "lifting lemmas") that if N is saturated with respect to B, then the set of its ground instances is also saturated. Therefore we may use the above lemma to infer that the relation $\downarrow_R$ is transitive and that I is a model of $N \cup T_<$. □

We can use this theorem to check the unsatisfiability of sets of Horn clauses over languages with transitive relations: saturate the given set of clauses with respect to the inference system B; if the set if unsatisfiable, a contradiction in the form of the empty clause will eventually be generated. In the following section we generalize the ordered chaining calculus B to arbitrary clauses. First, though, we describe a suitable concept of redundancy that is compatible with chaining.

3.4 Redundancy

We sketch the main ideas of an abstract notion of redundancy and refer to Bachmair and Ganzinger (1993c) for further details.

Let N and E be sets of clauses and C be a ground clause (not necessarily a ground instance of N). We call C *E-redundant* with respect to N if there exist ground instances $C_1, \ldots, C_k$ of N such that C is true in every E-model of $C_1, \ldots, C_k$ and $C \succ C_j$, for all j with $1 \leq j \leq k$. It can easily be seen that the clauses $C_1, \ldots, C_k$ can be assumed to be non-redundant. A non-ground clause is called E-redundant if all its ground instances are.

Tautologies are E-redundant in this sense, for any E, and most cases of proper subsumption are also covered by this notion of redundancy. The axioms in E are all E-redundant by definition (presumably they are built into the inference mechanism).

A ground inference with conclusion B and maximal premise C is called *E-redundant* with respect to N if either some premise is redundant, or else there exist ground instances $C_1, \ldots, C_k$ of N such that B is true in every E-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 E-redundant if all its ground instances are.

We say that a set of clauses N is *saturated up to E-redundancy* if all inferences from N are E-redundant.

For the remaining part of this section, redundancy is meant to refer to $T_<$-redundacy. One way to render an inference in B redundant is to add its conclusion to the set N.

3.5 Ordered chaining for general clauses

The inference system B is refutationally complete for Horn clauses, but not for general clauses. In particular, a factoring rule is needed for disjunctions of positive literals.

Ordered Factoring:

$$\frac{C \ , \ A \ , \ B}{C\sigma \ , \ A\sigma}$$

where σ is the most general unifier of A and B, and $A\sigma$ is maximal in $C\sigma$.

Unfortunately, it also turns out that for disjunctions of positive inequalities with multiple occurrences of the maximal term of a clause the inference mechanism of negative and ordered chaining is insufficient. For example, take the set of clauses

$$\begin{array}{rcl} & \rightarrow & c < b \\ & \rightarrow & b < c \\ & \rightarrow & a < b, a < c \\ a < b, a < c & \rightarrow & \end{array}$$

where $a \succ b \succ c$. This set of clauses is unsatisfiable. The third clause implies that $a < b$ or $a < c$ is true. But by the first two clauses, if one of the two inequalities is true, so is the other. This contradicts the last clause. From the first two clauses we obtain $c < c$ by chaining. From the last two clauses we obtain $b \not< b \vee a \not< c \vee a < c$ by negative chaining. This inference is redundant, as its conclusion is a tautology. Thus, by adding $c < c$ the set is saturated up to redundancy, but does not contain the empty clause.

The following inference rules restore refutational completeness and guarantee compatibility with redundancy.

Transitivity Resolution:

$$\frac{C\ ,\ u < v\ ,\ u' < v' \quad D\ ,\ s < t}{D\sigma\ ,\ t\sigma \not< v'\sigma\ ,\ s\sigma < v'\sigma}$$

where σ is the most general unifier of u, u' and s; the inequality $u\sigma < v\sigma$ is strictly maximal in $C\sigma \vee u'\sigma < v'\sigma$; the inequality $s\sigma < t\sigma$ is strictly maximal in $D\sigma$; and $v\sigma \not\preceq u\sigma$, $v'\sigma \not\preceq u\sigma$, $v'\sigma \not\preceq v\sigma$, $t\sigma \not\preceq s\sigma$, and $v\sigma \not\succ t\sigma$.

$$\frac{C\ ,\ u < v\ ,\ u' < v' \quad D\ ,\ s < t}{D\sigma\ ,\ u'\sigma \not< s\sigma\ ,\ u'\sigma < t\sigma}$$

where σ is the most general unifier of v, v' and t; the inequality $u\sigma < v\sigma$ is strictly maximal in $C\sigma \vee u'\sigma < v'\sigma$; the inequality $s\sigma < t\sigma$ is strictly maximal in $D\sigma$; and $u\sigma \not\preceq v\sigma$, $u'\sigma \not\preceq v\sigma$, $u'\sigma \not\preceq u\sigma$, $s\sigma \not\preceq t\sigma$, and $u\sigma \not\succ s\sigma$.

These inference rules represent the controlled application of resolution with the transitivity axiom. For instance, the clause $D\sigma \vee t\sigma \not< v'\sigma \vee s\sigma < v'\sigma$ is an instance of the conclusion of the resolution inference

$$\frac{D\sigma\ ,\ s\sigma < t\sigma \quad x \not< y\ ,\ y \not< z\ ,\ x < z}{D\sigma\ ,\ t\sigma \not< z\ ,\ s\sigma < z}$$

In other words, the conclusion of a transitivity resolution inference π is an instance of a resolvent between the second premise of π and the transitivity axiom. The first premise of π regulates which instances and resolvents are needed.

The calculus consisting of $\mathsf{B}_<$ plus ordered factoring and transitivity resolution is denoted by $\mathsf{C}_<$ (or C). These calculi are refutationally complete for arbitrary clause sets with a transitivity axiom. The completeness proof is based on a slightly modified model construction.

Let N be a set of ground clauses. For every clause C in N, let R_C be the set $\bigcup_{C \succ D} E_D$ and I_C be the rewrite closure $R_C \cup \{u < v : u \downarrow_{R_C} v\}$ of R_C. Furthermore, if C is a clause $C' \vee A$, where A is a positive literal and $A \succ C'$, and C is false in I_C and C' is false in the rewrite closure of $R_C \cup \{A\}$, then $E_C = \{A\}$. In all other cases, $E_C = \emptyset$. We denote by R the set $\bigcup_C E_C$ and by I the rewrite closure of R. As before, R^C denotes $R_C \cup E_C$ and I^C the rewrite closure of R^C.

Lemma 3 *If a ground clause C (which need not be in N) is true in some interpretation I_D or I^D, where $D \succeq C$, then C is also true in I and in any interpretation $I_{D'}$ and $I^{D'}$ with $D' \succ D$.*

Lemma 4 *Let N be a set of clauses that is saturated up to redundancy with respect to C and does not contain the empty clause, and let I be the interpretation constructed from all ground instances of N. Then for every ground instance C of a clause in N we have:*

(1) If C is productive then it is non-redundant.

(2) The relation $\to_{R^C}^{\equiv}$ commutes with $\leftarrow_{R^C}^{\equiv}$, and hence the relation $\downarrow_{R^C}^{\equiv}$ is transitive and I_C as well as I^C are transitivity interpretations.

(3) The clause C is true in the rewrite closure of R^C.

(4) If D is a productive clause $D' \vee A$, with $A \succ D'$ and if $C \succeq D$, then D' is false in the rewrite closure of R^C.

Proof. We use induction on $\succ$. Let C be a ground clause in N, such that (1)–(4) are satisfied for all smaller clauses in N.

(1) Let C be $T_<$-redundant in N. Then there exist ground instances $C_i, \ldots, C_n$ of N such that C logically follows from the C_i, that is, is true in every $T_<$-model of the C_i. Applying the induction hypothesis of (2) and (3) we infer that I_C is a $T_<$-model of the C_i. Therefore C is true in I_C and, hence, not productive.

(2) The commutation property is a consequence of saturation under ordered chaining. By the induction hypothesis, peaks that involve only R_C commute, hence I_C is a transitivity interpretation. The assertion for I^C is thus trivially true if E_C does not contain an inequality, or if $E_C = \{t < v\}$ where $v \succeq t$. Suppose $C = C' \vee v < t$ produces $v < t$ with $v \succ t$. By the definition of E_C, the clause C' is false in I^C. If there is another clause $D = D' \vee u < v$ that produces $u < v$ with $v \succ u$, then there exists a peak $u \leftarrow_{R_C} v \to_{E_C} t$. Since $C \succ D$ we may apply the induction hypothesis to conclude that D' is also false in I^C. Furthermore, C and D are by (1) non-redundant and produce $C'' = C' \vee D' \vee u < t$ by ordered chaining. Since N is saturated up to redundancy, the clause C'' must logically follow from clauses strictly smaller than C. That is, C'' must be true in I_C. Since $C' \vee D'$ is false in I_C, we conclude that $u < t$ is true in I_C, which establishes the required commutation property.

(3) We already know that all ground instances of N that are smaller than C are true in I_C and that I_C is a transitivity interpretation. If C is redundant, then it has to be true in I_C and in hence in I^C. If C is productive, it is by construction true in the rewrite closure of R^C. Let C be a non-redundant, non-productive clause.

(3.1) First suppose the maximal literal in C is negative and C can be written as $C' \vee \neg A$, with a maximal atom $\neg A$ that is false in I_C. Thus A is true in I_C.

If A is produced by some (smaller) clause $D = D' \vee A$, then the resolvent $C'' = D' \vee C'$ of C and D is smaller than C (since $\neg A \succ D'$). Since N is saturated up to redundancy, C'' must follow from clauses smaller than C, all of which are true in I_C. Therefore C'' must be true in I_C. By the induction hypothesis, D' is false in I_C. Therefore C', and also C, is true in I_C.

If A is an inequality $u < v$ with $u \downarrow_{R_C} v$, then there exists a suitable productive clause $D' \vee u < v'$ (where $v' \downarrow_{R_C} v$) or $D' \vee u' < v$ (where $u \downarrow_{R_C} u'$). By negative chaining we get either $C' \vee D' \vee v \not< v'$ or $C' \vee D' \vee u \not< u'$. In either case we may use saturation up to redundancy to infer that C', and hence C, is true in I_C.

(3.2) If C is of the form $C' \vee A \vee A$, with multiple occurrences of the maximal literal A, then the smaller clause $C' \vee A$ is obtained from C by ordered factoring. Using the induction hypothesis and saturation up to redundancy, we may infer that $C' \vee A$ and C must be true in I_C.

(3.3) Suppose C can be written as $C' \vee u < v \vee u < v'$, where $u \succ v \succ v'$ and $v < v'$ is true in I_C. Transitivity resolution of C with itself produces $C'' = C' \vee v \not< v' \vee u < v'$. Using the induction hypothesis and saturation up to redundancy, we may infer that C'' is true in I_C. Since $v \not< v'$ is false in I_C, this implies that $C' \vee u < v'$, and hence C, is true in I_C.

Similar arguments apply if C is of the form $C'' \vee u < v \vee u' < v$, where $v \succ u \succ u'$.

Cases (3.1)–(3.3) cover all the possibilities for a non-redundant clause C to be non-productive. In each case we have shown C to be true in I_C, which completes this case.

(4) Suppose D is a productive clause of the form $D' \vee A$ with $A \succ D'$ and $C \succeq D$. By means of inductive reasoning we may assume that D' is false in I_C. (This is easily confirmed for the base case $C = D$.) Suppose D' is true in the rewrite closure of R^C. There are essentially two cases. One is that C is of the form $C' \vee u < v$ and produces $u < v$ with $u \succ v$; and moreover A is an inquality $u < t$ and D' is of the form $D'' \vee u < t'$, such that $v < t'$ is true in I_C, but $t < t'$ is false in I_C. But then we may obtain the clause $C'' = C' \vee v \not< t' \vee u < t'$ by transitivity resolution from C and D. Since N is saturated up to redundancy, and all clauses smaller than C are true in I_C, we may infer that C'' is true in I_C. Also, since C is productive, C' is false in I_C. Therefore, $u < t'$ must be true in I_C. This contradicts our assumption that D' is false in I_C. The other, similar case that C is of the form $C' \vee u < v$ and produces $u < v$ with $v \succ u$, corresponds to the other version of transitivity resolution. Details are left to the reader. □

As an immediate corollary of the above lemma we obtain the following completeness theorem.

Theorem 2 *If a set of clauses N is saturated up to redundancy with respect to $\mathsf{C}_<$, then the set $N \cup T_<$ is unsatisfiable if and only if it contains the empty clause.*

3.6 Variable chaining

In chaining we put together two inequalities (one of which may be negative) by unifying the first argument of one with the second argument of the other. If one of the two terms thus unified is a variable we speak of a *variable chaining.* More specifically, we speak of chaining *into* a variable if the inequality with the variable is not smaller (with respect to $\succ$) than the other inequality; and speak of chaining *from* a variable if the inequality with the variable is not bigger than the other inequality. Since the unification of a term with a variable not occurring in it always succeeds, variable chaining can be particularly prolific. Fortunately, the combination of redundancy, which by Theorem 2 is compatible with chaining, and ordering constraints drastically cuts down on variable chaining.

Consider, for instance, an ordered chaining inference

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

where $x\sigma = t$. The constraints imposed on an ordered chaining inference require x to be a maximal term in the first premise, which implies that x does not occur in a negative literal and also is *unshielded* in the clause in the sense that it occurs only as an argument of $<$ but not as an argument of any other predicate symbol or any function symbol. Moreover, the inference is redundant if the variable x is *linear* in the first premise, i.e., does not occur in C or u. For in that case the conclusion is $C \vee D \vee u < v$, and is either properly subsumed by $C \vee u < x$ or else is identical to $C \vee u < x$ up to renaming of variables. Similar arguments apply to inferences

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

where the variable occurs as the first argument of an inequality.

In sum, ordered chaining through a variable is only necessary if the variable is non-linear and unshielded and occurs only in positive literals. For Horn clauses these conditions cannot be satisfied simultaneously, so that ordered chaining through variables is not needed at all. But negative chaining into non-linear unshielded variables is still necessary, even for Horn clauses, as the following example shows. Take the set of clauses

$$\begin{aligned} &\rightarrow \quad a < a' \\ &\rightarrow \quad a' < c \\ &\rightarrow \quad b < b' \\ &\rightarrow \quad b' < c \\ a < x \;,\; b < x \quad &\rightarrow \end{aligned}$$

where $c \succ b' \succ b \succ a' \succ a$. This set of clauses is unsatisfiable. From the first two clauses we obtain $a < c$; from the third and fourth clause $b < c$. The two inequalities contradict the last clause. No ordered chaining is possible. We may resolve the third with the last clause, to get $a \not< b'$, from which we get $a \not< b$ by chaining with the third clause again. Any other inferences require chaining through a variable. In other words, no contradiction can be derived if we exclude variable chaining. If we use the fourth and fifth clause to chain through a variable, we get $a \not< c \vee b \not< b'$, from which a contradiction can be derived by one more chaining and two resolution steps.

In the next section we show that chaining into variables can be avoided if the given transitive relation is also symmetric.

4 Partial Equivalence Relations

Let $\sim$ be a transitive relation that also satisfies the symmetry axiom

$$x \sim y \rightarrow y \sim x.$$

This clause can be resolved with every clause $C \vee s \sim t$ for which $s \sim t$ is a strictly maximal literal, to yield $C \vee t \sim s$. Thus, symmetry is usually built into chaining by using literals $s \sim t$ in both directions.

Consider now a chaining inference

$$\frac{C \;,\; s \sim t \quad D \;,\; u \not\sim x}{C \;,\; D\sigma \;,\; u\sigma \not\sim t}$$

where $x\sigma = s$, $t \not\succeq s$, $s \sim t$ is strictly maximal in C, and $u\sigma \not\sim s$ is maximal in $D\sigma$. Let τ be a substitution with $x\tau = t$. Then the conclusion of the chaining inference follows from $D\tau \vee u\tau \not\sim t$ and $C \vee s \sim t$, and the inference can easily be seen to be redundant. Thus, chaining into a variable in a negative literal is unnecessary. (Note that this holds for clauses in general, not just Horn clauses.) As a consequence, the resolution inferences with the symmetry axiom mentioned above are the only inferences in which this axiom can participate. Thus, the presence of the symmetry axiom allows us to restrict variable chainings, which in turn cuts down on the number of inferences with symmetry.

Similar arguments as above apply to most[7] chainings into a variable in a positive inequality. Chaining *from* a variable (in a positive literal) is still needed, though. For

[7] namely those which are not also chainings from a variable

example, the set of two clauses

$$\begin{array}{rcl} & \rightarrow & x \sim a \\ a \sim b & \rightarrow & \end{array}$$

is unsatisfiable. But if $c \succ b \succ a$ (and because of the minimality of the constant a, also $x \succeq a$), then we need to chain through x in the first clause to derive a contradiction. (If a is not minimal, we can chain the first clause with itself through a, to obtain $x \sim y$, which can then be resolved with the second clause to yield a contradiction.)

An important consequence of symmetry is that transitivity resolution can be restricted. More precisely, let $\mathsf{S}_\sim$ be the inference system $\mathsf{C}_\sim$, but with the inference rules modified so that symmetry of $\sim$ is taken into account and with the transitivity resolution rules replaced by the following inference rule.

Equality Factoring:

$$\frac{C \ , u \sim v \ , u' \sim v'}{C\sigma \ , v\sigma \not\sim v'\sigma \ , u\sigma \sim v'\sigma}$$

where σ is the most general unifier of u and u'; $v\sigma \not\succeq u\sigma$, $v'\sigma \not\succeq u\sigma$, and $v'\sigma \not\succeq v\sigma$; and $u\sigma \sim v\sigma$ is strictly maximal in $C\sigma \vee u'\sigma \sim v'\sigma$.

Equality factoring corresponds to transitivity resolution of a (suitable) clause with itself. Any other transitivity resolutions are unnecessary for symmetric relations. More precisely, Lemma 4 can be strengthened as outlined below.

We first modify the model construction to take account of symmetry. For any ground atom A, we define A^{-1} as follows. If A is an atom $u \sim v$, let A^{-1} be $v \sim u$; otherwise let A^{-1} be A. Let N be a set of ground clauses. The sets R_C, R^C, I_C, I^C, R, and I are defined as before. However, if C is a clause $C' \vee A$, where A is a positive literal and $A \succ C'$, and C is false in I_C and C' is false in the rewrite closure of $R_C \cup \{A\}$, then $E_C = \{A, A^{-1}\}$. Thus rewrite rules in R_C can be used in both directions (which is reflected in the inference rules of S).

Let us denote by $ST_\sim$ the set consisting of the symmetry and transitivity axioms for the predicate $\sim$.

Lemma 5 *Let N be a set of clauses that is saturated up to $ST_\sim$-redundancy with respect to S and does not contain the empty clause, and let I be the interpretation constructed from all ground instances of N. Then for every ground instance C of a clause in N we have:*

(1) if C is productive then it is non-redundant.

(2) The relation $\rightarrow^{\equiv}_{R^C}$ commutes with $\leftarrow^{\equiv}_{R^C}$, and hence the relation $\downarrow^{\equiv}_{R^C}$ is transitive. Furthermore, the interpretations I_C and I^C satisfy $ST_\sim$.

(3) The clause C is true in I^C.

(4) If C is a productive clause $C' \vee u \sim v$ (or $C' \vee v \sim u$) with $u \succ v$, and D is a productive clause $D' \vee s \sim t$ (or $D' \vee t \sim s$) with $s \succ t$, and if in addition $C \succ D$, then the terms u and s are different.

Proof. The proof is similar to the proof of Lemma 4, and parts (1)–(3) are essentially the same.

(4) Let C and D be productive, and hence non-redundant, clauses as indicated, and suppose s and u are the same. Then we may obtain the clause $C'' = C' \vee D' \vee t \sim v$

by ordered chaining. Using saturation up to redundancy and the induction hypothesis, we may infer that C'' is true in I_C, whereas $C' \vee D'$ is false in I_C. Consequently, $t \sim v$ must be true in I_C. But since $s \sim t$ is also true in I_C and I_C satisfies $ST_\sim$, we conclude that $s \sim v$ is true in I_C, which contradicts the assumption that C is productive. □

This lemma does indeed strengthen Lemma 4. For if $C = C' \vee A$ is productive, then C' is false in I, as any atom in $I \setminus I_C$ can only be used to rewrite terms bigger than the maximal term in C. The lemma also indicates that a ground term can be rewritten by at most one rule in R. In other words, rewrite proofs are "deterministic" on the ground level, which may be of advantage in checking for redundancy.

Let us conclude this section with a remark on *equivalence relations*, which are transitive symmetric relations that are also reflexive, a property that can be built in as an (harmless) inference:

Reflexivity Resolution:

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

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

5 Congruence Relations

Rewrite techniques have primarily been applied to equivalence relations $\sim$ that also satisfy the congruence axioms

$$\begin{array}{rcl} x \approx y & \rightarrow & f(\ldots, x, \ldots) \approx f(\ldots, y, \ldots) \\ x \approx y \, , P(\ldots, x, \ldots) & \rightarrow & P(\ldots, y, \ldots) \end{array}$$

for all function symbols f and predicate symbols P. (For each symbol of arity n, we need n axioms.)

Repeated resolutions with congruence axioms allow one to build a "context" around the terms in a maximal positive literal, so that from a clause $C \vee s \approx t$ with maximal "equality" literal $s \approx t$ we may produce any clause $C \vee u[s] \approx u[t]$. Combining such inferences with chaining, we obtain what may be called subterm chaining, but is actually known as paramodulation or superposition. (Paramodulation corresponds to ordinary chaining, while superposition also includes ordering restrictions.) A typical example is the following inference rule.

Superposition:

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

where (i) σ is a most general unifier of s and s', (ii) $s\sigma \approx t\sigma$ is strictly maximal in $C\sigma$, and $t\sigma \not\succeq s\sigma$, (iii) $u\sigma \approx v\sigma$ is strictly maximal in $D\sigma$, and $v\sigma \not\succeq u\sigma$, and (iv) s' is not a variable.

A similar inference rule is needed for superposition into negative equalities. The congruence axioms for predicate symbols also require one to paramodulate into arbitrary literals.

Ordered Paramodulation:

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

where (i) σ is a most general unifier of s and s', (ii) $s\sigma \approx t\sigma$ is strictly maximal in $C\sigma$, and $t\sigma \not\succeq s\sigma$, (iii) $L[s']\sigma$ is maximal in $D\sigma$, and (iv) s' is not a variable.

Paramodulation was introduced by Robinson and Wos (1969), but all early completeness proofs assumed the presence of "functional-reflexive axioms" and required paramodulation into variables (i.e., chaining into variables). Brand (1975) was the first to prove that the functional-reflexive axioms are not needed, and his proof requires only a very limited form of paramodulation into a variable. The first proof that paramodulation into a variable is not needed at all was given by Peterson (1983), while Hsiang and Rusinowitch (1991) were the first to explicitly put some ordering restrictions on paramodulation. The more restrictive superposition calculus described in Zhang and Kapur (1988), Zhang (1988) turns out to be incomplete; for a counterexample see Bachmair and Ganzinger (1990). The superposition calculus with equality factoring was introduced by Bachmair and Ganzinger (1990). There are also alternatives to equality factoring: Rusinowitch (1991) weakens some ordering constraints, while merging paramodulation is proposed in Bachmair and Ganzinger (1990), see also Pais and Peterson (1991). More recently new paramodulation and superposition calculi have been proposed by Bachmair, Ganzinger, Lynch, et al. (1992) and Nieuwenhuis and Rubio (1992).

We emphasize that the above results apply to *symmetric* relations. There are interesting examples of theories with non-symmetric operators that satisfy congruence or similar properties. For example, to define ring structures we may need axioms, such as

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

that express monotonicity or anti-monotonicity properties of a binary relation with respect to certain functions. Naturally, (subterm) chaining methods for general clauses (Manna and Waldinger 1986, Manna and Waldinger 1992) and completion-like procedures for unit clauses (Levy and Agustí 1993) have been proposed for such relations, but for completeness chaining into the variables of functional-reflexive axioms (cf. the "variable instance pairs" in Bachmair, Dershowitz and Hsiang 1986) is required, which is impractical in general.

For example, given two inequalities $x < x * x$ and $a < b$, and a syntactic ordering in which $a \succ b$ and $x * x \succ x$, we need the "variable subterm chaining" (or "variable overlap") $a < a * a < a * b$ to derive $a < a * b$. (If $<$ were symmetric, we would also have $a < b < b * b < a * b$ and the chaining would be unnecessary.) This example, which is typical of the problems variable overlaps may cause for completion procedures, depends on the syntactic ordering $\succ$ and on the non-linearity of the variable x. In the case of non-unit clauses, proper handling of negative literals poses further difficulties.

Take the set of two clauses

$$\begin{aligned} &\rightarrow \quad a < b \\ f(x) < f(f(y)) \quad &\rightarrow \end{aligned}$$

(and the monotonicity axiom $x < y \rightarrow f(x) < f(y)$. This set of clauses is unsatisfiable: from the first clause and the congruence axiom we get $f(f(a)) < f(f(b))$, which contradicts the second clause. However, we cannot get a contradiction from the first two clauses by chaining even at variable subterms (regardless of how the ordering is defined). In this example we can get a contradiction by an inference that combines chaining into a variable with a subsequent unification of terms. Thus, we get $f(x) < f(f(a))$ by chaining, and the empty clause, because $f(x)$ and $f(f(a))$ are unifiable. This is a limited form of resolution, which may be called *context resolution*, in which a context is put around one of the literals before the actual resolution takes place. In the example above, the context is determined by the chaining and the subsequent unification. Unfortunately, in the presence of non-linear variables, the context may have to be guessed. Take the clauses $a < b$ and $f(g(b), x, x) \not< f(y, y, g(a))$. Assuming suitable congruence axioms, the first clause implies $f(g(b), g(a), g(a)) < f(g(b), g(b), g(a)$, which indicates the appropriate context for $a < b$, so that resolution with the second clauses yields a contradiction.

6 Summary

We have proposed chaining with ordering restrictions as an inference mechanism for dealing with transitive relations and have established its refutational completeness for Horn clauses. We have also presented an extension to the inference system for Horn clauses that renders it refutationally complete for general clauses and, in the case of symmetric transitive relations, reduces to equality factoring, which is known from superposition calculi. All inference systems are compatible with a notion of redundancy that covers the simplification techniques commonly used in rewrite-based theorem provers.

In this paper we have discussed transitive relations, partial equivalence relations, and congruence relations. But our approach and methodology are also applicable to other theories, of which perhaps total orderings, satisfying the axiom

$$x < y \ , \ y < x \ , \ x \approx y$$

are of particular interest, as they require the consideration of a combination of two transitive relations—equality and inequality—and their corresponding commutation properties. A related topic is the integration of equational theories such as associativity and commutativity in, say, a superposition calculus, where one also has to deal with the combination of two equational theories and corresponding commutation properties, see Bachmair and Ganzinger (1993a).

Chaining techniques have also been applied to set theory Hines (1990), though no completeness results are known yet for the theorem proving systems that have been proposed. Other applications include the application of standard theorem proving techniques to certain modal and temporal logics based on their Kripke semantics. The accessibility relations on which these semantics are based are usually transitive (see Ohlbach 1993 for an overview of translation methods for such logics). In these logics the entailment is still monotone so that it can be identified with entailment in first-order logic. It is conceivable that ordered chaining can also be applied fruitfully to certain logics for which entailment is non-monotone but transitive.

A key question in all applications of chaining, also from a practical point of view, is to what extent variable chaining is needed. The ordering restrictions imposed in our calculi considerably cut down on such chainings, though some are still needed. In the presence of symmetry, chaining into variables is not needed. Variable chainings can be completely excluded for dense total orderings with no endpoints, provided certain variable elimination rules are applied (Bledsoe and Hines 1980, Bledsoe, Kunen and Shostak 1985, Hines 1992). The application of rewrite techniques to such theories is discussed in Bachmair and Ganzinger (1993b).

References

L. Bachmair and N. Dershowitz, 1986. Commutation, transformation, and termination. In J. H. Siekmann, editor, *Proc. 8th Int. Conf. on Automated Deduction*, Lecture Notes in Computer Science, vol. 230, pp. 5–20, Berlin, Springer-Verlag.

L. Bachmair, N. Dershowitz and J. Hsiang, 1986. Orderings for equational proofs. In *Proc. Symp. Logic in Computer Science*, pp. 346–357, Boston, Massachusetts.

L. Bachmair, N. Dershowitz and D. Plaisted, 1989. Completion without failure. In H. Ait-Kaci, M. Nivat, editors, *Resolution of Equations in Algebraic Structures, vol. 2*, pp. 1–30. Academic Press.

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. Associative-Commutative Superposition. Technical Report MPI-I-93-250, Max-Planck-Institut für Informatik, Saarbrücken.

L. Bachmair and H. Ganzinger, 1993. Ordered Chaining for Total Orderings. Technical Report MPI-I-93-251, Max-Planck-Institut für Informatik, Saarbrücken.

L. Bachmair and H. Ganzinger, 1993. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation.* To appear.

L. Bachmair, H. Ganzinger, Chr. Lynch and W. Snyder, 1992. Basic Paramodulation and Superposition. In D. Kapur, editor, *Automated Deduction — CADE'11*, Lecture Notes in Computer Science, vol. 607, pp. 462–476, Berlin, Springer-Verlag.

F. Bellegarde and P. Lescanne, 1987. Transformation Ordering. In H. Ehrig, R. Kowalksi, G. Levi, U. Montanari, editors, *TAPSOFT'87*, volume 249, pp. 69–80.

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.

N. Dershowitz, 1991. A Maximal-Literal Unit Strategy for Horn Clauses. In *Proc. 2nd Int. Workshop on Conditional and Typed Rewriting Systems*, Lecture Notes in Computer Science, vol. 516, pp. 14–25, Berlin, Springer-Verlag.

N. Dershowitz and J.-P. Jouannaud, 1990. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science B: Formal Methods and Semantics*, chapter 6, pp. 243–309. North-Holland, Amsterdam.

H. Ganzinger, 1991. A completion procedure for conditional equations. *Journal of Symbolic Computation*, Vol. 11, pp. 51–81.

A. Geser, 1991. Relative Termination. Ulmer Informatik–Berichte 91-03, Universiät Ulm, Fakultät für Informatik, Ulm, FRG.

L. Hines, 1990. Str+ve⊆: The Str+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. W. Klop, 1992. Term Rewriting Systems. In S. Abramsky, D.M. Gabbay and T.S.E. Maibaum, editors*Handbook of Logic in Computer Science*, volume 2, pp. 2–116. Clarendon Press, Oxford.

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.

Z. Manna and R. Waldinger, 1986. Special Relations in Automated Deduction. *Journal of the ACM*, Vol. 33, pp. 1–60.

Z. Manna and R. Waldinger, 1992. The Special-Relation Rules are Incomplete. In Deepak Kapur, editor, *11th International Conference on Automated Deduction*, pp. 492–506, Saratoga Springs, NY, Springer LNAI 607.

R. Nieuwenhuis and A. Rubio, 1992. Basic superposition is complete. In *ESOP'92*, Lecture Notes in Computer Science, vol. 582, pp. 371–389, Berlin, Springer-Verlag.

H. J. Ohlbach, 1993. Translation Methods for Non-Classical Logics – An Overview. Technical Report MPI-I-93-225, Max-Planck-Institut für Informatik, Saarbrücken.

John Pais and G.E. Peterson, 1991. Using Forcing to Prove Completeness of Resolution and Paramodulation. *Journal of Symbolic Computation*, Vol. 11, pp. 3–19.

Gerald E. Peterson, 1983. A technique for establishing completeness results in theorem proving with equality. *SIAM J. on Computing*, Vol. 12, No. 1, pp. 82–100.

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.

M. Rusinowitch, 1991. Theorem proving with resolution and superposition: An extension of the Knuth and Bendix completion procedure as a complete set of inference rules. *J. Symbolic Computation*, Vol. 11, pp. 21–49.

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.

H. Zhang, 1988. *Reduction, superposition and induction: Automated reasoning in an equational logic*. PhD thesis, Rensselaer Polytechnic Institute, Schenectady, New York.

H. Zhang and D. Kapur, 1988. First-order theorem proving using conditional rewrite rules. In E. Lusk, R. Overbeek, editors, *Proc. 9th Int. Conf. on Automated Deduction*, Lecture Notes in Computer Science, vol. 310, pp. 1–20, Berlin, Springer-Verlag.