# Ordered Chaining Calculi for First-Order Theories of Transitive Relations

Leo Bachmair
SUNY at Stony Brook, Stony Brook, New York
and
Harald Ganzinger
Max-Planck-Institut für Informatik, Saarbrücken, Germany

We propose inference systems for binary relations that satisfy composition laws such as transitivity. Our inference mechanisms are based on standard techniques from term rewriting and represent a refinement of chaining methods as they are used in the context of resolution-type theorem proving. We establish the refutational completeness of these calculi and prove that our methods are compatible with the usual simplification techniques employed in refutational theorem provers, such as subsumption or tautology deletion. Various optimizations of the basic chaining calculus will be discussed for theories with equality and for total orderings. A key to the practicality of chaining methods is the extent to which so-called variable chaining can be avoided. We demonstrate that rewrite techniques considerably restrict variable chaining and that further restrictions are possible if the transitive relation under consideration satisfies additional properties, such as symmetry. But we also show that variable chaining cannot be completely avoided in general.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*computational logic, mechanical theorem proving, proof theory*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*deduction, inference engines, resolution*

General Terms: Theory

Preliminary versions of the results in this paper have been presented at the *12th International Conference on Automated Deduction* (Nancy, France, June/July 1994) and the *9th IEEE Symposium on Logic in Computer Science* (Paris, France, July 1994).
This research was supported in part by the National Science Foundation under grants INT-9314412 and CCR-9510072, 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 would like to acknowledge support from the Alexander von Humboldt Foundation.
Name: Leo Bachmair
Affiliation: Department of Computer Science, SUNY at Stony Brook, Stony Brook, New York
Address: Stony Brook, NY 11794, email: `leo@cs.sunysb.edu`
Name: Harald Ganzinger
Affiliation: Max-Planck-Institut für Informatik, Saarbrücken, Germany
Address: Im Stadtwald, D-66123 Saarbrücken, Germany, email: `hg@mpi-sb.mpg.de`

Additional Key Words and Phrases: Chaining calculi, equational logic, reduction orderings, rewrite systems, term rewriting, transitive relations

## 1. INTRODUCTION

Resolution, in its different variants, forms the core of many current automated reasoning systems, see Bachmair and Ganzinger [1997] for a recent exposition of the theory of resolution. By and large such resolution refinements as hyper-resolution, ordered resolution, or the set-of-support strategy are useful in practice, but for theories with transitive relations, such as logics with equality or inequality relations, they are often not very effective. Special techniques have been devised for such theories: chaining and variable elimination for inequalities, and paramodulation and superposition for equality. Chaining essentially encodes certain resolution steps with the transitivity axiom. For example, if $S$ is a transitive relation, chaining allows one to derive $u\sigma\, S\, v\sigma$ from $u\, S\, s$ and $t\, S\, v$, where $\sigma$ is a most general unifier of $s$ and $t$. Corresponding calculi have been described by Bledsoe, Kunen, and Shostak [1985] and Hines [1992].

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. Unrestricted chaining reflects a notion of proof in which the transitivity property is captured by a sequence of term replacement steps, whereas ordered chaining represents a restriction to so-called rewrite proofs, in which replacement steps are constrained by a term ordering. The connection between arbitrary proofs and rewrite proofs is established via commutation properties of underlying binary relations. Intuitively, commutation guarantees that the transitive closure of a relation coincides with the "rewrite closure," thereby justifying the restriction to rewrite proofs.

Ordered chaining is an application of standard techniques from term rewriting, as exemplified by completion procedures. Commutation properties such as confluence provide the theoretical foundation of completion, though in that case rewrite closures are defined via *subterm* replacement and only for congruence relations. In this paper, on the other hand, we investigate the application of rewrite techniques to arbitrary (e.g., non-symmetric) transitive relations and design corresponding inference systems for first-order clauses, and not just unit clauses as is the case with standard completion procedures. The integration of rewrite techniques is relatively straightforward for Horn clauses, but more difficult in the presence of disjunctions of positive literals, due to the additional degree of non-determinism in the rewrite relations and certain subtle dependencies between them. Ordered inference systems allow one to effectively restrict deductive search and forward reasoning by simplification via don't-care nondeterministic backward reasoning. These simplification techniques are indispensable for the practical performance of automated theorem provers. They are based on a suitable notion of redundancy.

The practicality of chaining methods also 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 transitive predicate $S$, but not as an argument of a function symbol. However, we also show that some variable chainings are needed for transitive relations in general, though further restrictions 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 main ideas of our approach are introduced in Section 3, in a way that does not require too much technical detail, for the simple case of Horn clauses. Transitivity is an instance of a composition law and ordered chaining techniques are applicable to a large class of such composition laws, as described in Section 4, where we also consider full first-order clauses and outline a notion of redundancy for chaining calculi that covers the usual simplification techniques used in refutational theorem provers. Refutational completeness is established for an appropriately extended inference system. Redundancy is a central concept when one wants to tailor the general inference system to interesting special cases of transitive relations, so as to further optimize the inference rules and avoid variable chaining as much as possible. In particular, we investigate congruence relations, in Section 5, and total orderings, in Section 6.

## 2. PRELIMINARIES

We consider first-order languages with function and 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 $n$, $t_1, \ldots, t_n$ are terms, and $x$ is a variable. Function symbols of arity 0 are called *constants*. We assume that a language contains at least one constant, so that the set of variable-free terms is non-empty. 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. We often use infix notation for binary predicates and write, for instance, $u\, S\, v$ instead of $S(u, v)$. An expression is said to be *ground* if it contains no variables.

A *clause* is a finite multiset of literals (representing a disjunction). 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 $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$. One advantage of representing clauses as multisets of literals is that an ordering on literals can easily be extended to clauses. The multiset extension $\succ_{mul}$ of an ordering $\succ$ is defined by: $M \succ_{mul} N$ if and only if $M \neq N$ and $N$ can be obtained from $M$ by replacing some elements by zero or more smaller elements. For simplicity, we usually use the same symbol $\succ$ to denote both an ordering and its multiset extension.

By an (*Herbrand*) *interpretation* we mean a set $I$ of ground atoms. We say that a ground 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 $A \notin I$. If $P$ is an $n$-ary predicate, we denote by $P_I$ the $n$-ary relation $\{(t_1, \ldots, t_n) : P\,(t_1, \ldots, t_n) \in I\}$ on ground terms.

A ground clause is *true* in an interpretation $I$ if at least one of its literals is true in $I$; and *false* otherwise. A non-ground clause is said to be true in $I$ if all its ground instances are true. (An instance of an expression $E$ is obtained by substituting terms for variables; a ground instance, by substituting ground terms. By $E\sigma$ we denote the result of applying a substitution $\sigma$ to an expression $E$.) The *empty* *clause* (which is represented by the empty multiset) is false in every interpretation. We say that $I$ is a *(Herbrand) 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* or an $N$-*model*. A set of clauses is called *satisfiable* if it has a model, and *unsatisfiable* otherwise. For instance, any set containing the empty clause is unsatisfiable.

Refutational theorem provers attempt to show that a formula is valid by deriving a contradiction from its negation. Input formulas are typically preprocessed before new formulas are deduced from them. In particular, quantifiers are eliminated via Skolemization and formulas are translated to clause form—a reformulation that reflects a view of the theorem proving process as a method of checking the unsatisfiability of formulas over Herbrand interpretations. We will discuss techniques for dealing with transitive binary relations from this perspective.

## 3. CHAINING INFERENCES FOR TRANSITIVE RELATIONS

If $S$ is a binary predicate symbol, the corresponding *transitivity law* for $S$ can be written as a clause

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

An interpretation that satisfies this transitivity law is also called a *transitivity interpretation* for $S$. Evidently, if $I$ is a transitivity interpretation for $S$, then the relation $S_I$ is transitive.

An explicit representation of transitivity laws as clauses presents difficulties for traditional resolution-based provers. A transitivity law can be resolved with (a variant of) itself to yield a new clause

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

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

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

and so on. In other words, a resolution prover may derive a potentially infinite set of clauses to express that from any finite sequence of facts

$$u_0\,S\,u_1, u_1\,S\,u_2, \ldots, u_{n-1}\,S\,u_n$$

one may deduce $u_0\,S\,u_n$ whenever $S$ denotes a transitive predicate.

Certain refinements of resolution provide less prolific inference mechanisms for dealing with transitivity. For instance, problematic self-resolutions can be avoided by a judicious use of selection functions [Bachmair and Ganzinger 1994]: if one selects the two negative literals in the transitivity law, then the only inferences involving transitivity are of the form

$$\frac{C\,,\,u\,S\,s \quad D\,,\,t\,S\,v \quad \neg(x\,S\,y)\,,\,\neg(y\,S\,z)\,,\,x\,S\,z}{C\sigma\,,\,D\sigma\,,\,u\sigma\,S\,v\sigma}$$

where $\sigma$ is a most general unifier of $s$ and $t$. The key observation here is that it is not necessary to resolve with the non-selected, positive literals; see Bachmair and Ganzinger [1997] for details. In essence this amounts to an encoding of transitivity as an inference rule:

*Chaining.*

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

where $\sigma$ is a most general unifier of $s$ and $t$.

This chaining rule, which was already proposed by Slagle [1972], is not completely satisfactory either, as it can be used to explicitly generate the transitive closure of a binary relation. For example, from a single unit clause $x\,S\,fx$, infinitely many new clauses

$$x\,S\,ffx, x\,S\,fffx, x\,S\,ffffx, \ldots$$

can be derived by chaining.[1]

The point is that in refutational theorem proving, where the goal is to derive a refutation, only finitely many of these consequences of transitivity are needed. (After all a refutation is a *finite* derivation of a contradiction.) Our thesis is that term rewriting techniques may serve as a suitable general mechanism for controlling inferences with transitivity laws. The basic idea is that rewrite systems—sets of rewrite rules—allow for a less direct, but more compact representation of transitive closures. More specifically, we will interpret atomic formulas $u\,S\,v$ as rewrite rules and establish a suitable connection between the transitive closure of the relation denoted by $S$ and the rewrite closure induced by a corresponding set of rewrite rules. This approach allows us to formulate and justify various refinements of chaining that provide a basis for practical deductive systems. Rewrite techniques have traditionally been applied to equivalence or congruence relations only. Our approach to rewriting for non-symmetric relations is similar to what has been called bi-rewriting by Levy and Agustí [1993]. It can be used in more general contexts than just transitivity laws.

### 3.1 Composition laws

Transitivity may be viewed as a special case of a *composition law*

$$x\,S\,y\,,\,y\,T\,z \rightarrow x\,U\,z$$

that expresses a subset relation between binary relations: if an interpretation $I$ is a model of the composition law, then $S_I \circ T_I \subseteq U_I$, where $\circ$ denotes composition of relations. We will design inference rules for dealing with certain collections of composition laws.

Let $\mathsf{P}$ be a set of binary predicates. A strict partial order $<_{\mathsf{P}}$ on the set $\mathsf{P}$ is called *admissible* if (i) it is well-founded and (ii) for each symbol $S$ in $\mathsf{P}$, the set of all symbols smaller than $S$ is totally ordered. For instance, if $\mathsf{P}$ consists of three symbols $S$, $T$ and $U$, then an ordering with $S <_{\mathsf{P}} T$ and $S <_{\mathsf{P}} U$ is admissible, whereas an ordering with $T <_{\mathsf{P}} S$ and $U <_{\mathsf{P}} S$ is not.

Any admissible ordering $<_{\mathsf{P}}$ implicitly specifies a set of composition laws. More precisely, we denote by $\mathsf{CL}_{\mathsf{P}}$ the set of all clauses

$$x\,S\,y\,,\,y\,T\,z \rightarrow x\,U\,z,$$

[1] For simplicity, we omit parentheses for unary function symbols $f$ from terms.

where $S$ and $T$ are comparable (i.e., either $S \leq_{\mathsf{P}} T$ or $T \leq_{\mathsf{P}} S$) and $U$ is the maximum of $S$ and $T$. The symbols in $\mathsf{P}$ are also called *special predicates.*

For example, if $\mathsf{P}$ consists of three predicate symbols $\approx$, $\leq$, and $<$ (listed in increasing order), then $\mathsf{CL_P}$ consists of the following clauses:

$$
\begin{aligned}
x < y\,, y < z &\;\rightarrow\; x < z \\
x \approx y\,, y \approx z &\;\rightarrow\; x \approx z \\
x \leq y\,, y \leq z &\;\rightarrow\; x \leq z \\
x \approx y\,, y < z &\;\rightarrow\; x < z \\
x < y\,, y \approx z &\;\rightarrow\; x < z \\
x \approx y\,, y \leq z &\;\rightarrow\; x \leq z \\
x \leq y\,, y \approx z &\;\rightarrow\; x \leq z \\
x < y\,, y \leq z &\;\rightarrow\; x < z \\
x \leq y\,, y < z &\;\rightarrow\; x < z
\end{aligned}
$$

Note that the set $\mathsf{Tr_P}$ of all transitivity laws for symbols in $\mathsf{P}$ is a subset of $\mathsf{CL_P}$. In other words all special predicates denote transitive relations. The two sets $\mathsf{Tr_P}$ and $\mathsf{CL_P}$ are identical if $<_{\mathsf{P}}$ is the trivial (i.e., empty) ordering, in which no symbols are comparable.

### 3.2 Rewrite closures

We will represent relations satisfying composition laws by "composition closures" and examine the connection with rewrite closures.

A *chain* (with respect to $\mathsf{P}$) is a finite sequence of triples

$$(u_0, S_1, u_1), (u_1, S_2, u_2), \ldots, (u_{n-1}, S_n, u_n)$$

where $n \geq 1$, all terms $u_0, \ldots, u_n$ are ground and the set $\{S_1, \ldots, S_n\}$ is totally ordered. The *type* of a chain is the atom $u_0\, S_i\, u_n$, where $S_i$ is the maximal symbol among $S_1, \ldots, S_n$. We often use an abbreviated notion and write a chain as a sequence

$$u_0\, S_1\, u_1\, S_2\, u_2 \cdots S_n\, u_n.$$

For example, $a < fa \approx fb$ is a chain of type $a < fb$.

Chains represent derivations based on composition laws. A chain is called a *proof* in a Herbrand interpretation $I$, if all atoms $u_{i-1}\, S_i\, u_i$ are true in $I$. We also say that $u_0\, S\, u_n$ is *provable* in $I$, if there exists a proof of type $u_0\, S\, u_n$ in $I$. This notion of a proof has been defined in such a way that (i) [subproof property] any subchain $(u_{i-1}, S_1, u_i), \ldots, (u_{j-1}, S_n, u_j)$ of a proof is itself a proof, and (ii) [subproof replacement property] if any subchain is replaced by a proof (of the same type), the result is again a proof (of the same type).[2]

The *composition closure* of $I$ (with respect to $\mathsf{CL_P}$) is defined as the set $I$ plus all ground atoms $u\, S\, v$ that are provable in $I$.[3] In the special case of a set $\mathsf{Tr_P}$ we

[2] Since the ordering $<_{\mathsf{P}}$ is admissible, the set of predicate symbols occurring in any proof resulting from a subproof replacement is again totally ordered.

[3] Note that $I$ may be an interpretation over a language that also contains non-special predicate symbols.

also speak of a *transitive closure*. The following assertion can be easily proved.

PROPOSITION 1. *A Herbrand interpretation $I$ is a model of a set of composition laws $\mathsf{CL_P}$ if and only if it is identical to its composition closure (with respect to $\mathsf{CL_P}$).*

PROOF. For one direction, use induction on the length of proofs to show that whenever $I$ is a model of $\mathsf{CL_P}$, then it is identical to its composition closure. The other direction is straightforward. $\square$

A key technique in term rewriting is to consider only proofs of a certain form, typically called rewrite proofs, and to characterize under which conditions provability in the restricted sense coincides with provability in general. We next outline the basic concepts of this rewriting approach, in so far as they are relevant to chaining calculi.

Rewrite proofs are defined with respect to a total well-founded ordering $\succ$ on ground terms. We first classify each *proof step* $(u, S, v)$ according to the order relation between the two terms. We speak of a *(left-to-right) rewrite step* and write $u \Rightarrow_S v$, if $u \succ v$; speak of a *(right-to-left) rewrite step* and write $u \Leftarrow_S v$, if $v \succ u$; and speak of an *identity step* and write $u \Leftrightarrow_S v$, if $u = v$. (Totality and well-foundedness of the ordering $\succ$ ensure that each proof step will fall into exactly one of the three categories.) The annotation $S$ will be deleted if it is clear from the context or inessential. We then classify chains according to their individual steps. By a *valley* we mean a chain of the form

$$u_0 \Rightarrow u_1 \ \cdots \ \Rightarrow u_k \Leftarrow \ u_{k+1} \Leftarrow \cdots \ \Leftarrow u_n$$

or

$$u_0 \Rightarrow u_1 \ \cdots \ \Rightarrow u_k \Leftrightarrow \ u_{k+1} \Leftarrow \cdots \ \Leftarrow u_n.$$

Valleys are also called *rewrite proofs*. They consists of a sequence of left-to-right rewrite steps followed by at most one identity step and a sequence of right-to-left rewrite steps. A two-step chain $u \Leftarrow t \Rightarrow v$ is called a *peak*; a chain $u \Leftrightarrow u \Rightarrow v$ or $u \Leftarrow v \Leftrightarrow v$, a *plateau.* A chain $u = u_0 \Leftrightarrow u_1 \ \cdots \ \Leftrightarrow u_k = v$ is called a *plain* if it contains at least two steps (i.e., $k \geq 2$). A valley, as one can easily see, is a chain that contains no peak, plateau or plain.

We write $u \Downarrow_S^I v$, or simply $u \Downarrow_S v$, to indicate that there exists a rewrite proof of (type) $u\,S\,v$ in $I$. We say that a peak, plateau or plain *commutes* in $I$ if there exists a rewrite proof of the same type in $I$. The *rewrite closure* of $I$ is defined as the set $I$ plus all ground atoms that have a rewrite proof in $I$, that is, the set $I \cup \{u\,S\,v : u \Downarrow_S^I v\}$.

The rewrite closure of $I$ is obviously contained in the composition closure of $I$. We next characterize under which conditions any atom that is provable also has a rewrite proof, and for that purpose define a well-founded ordering on chains by associating a certain *complexity* with each proof step. More specifically, the complexity of a rewrite step $u \Rightarrow v$ is defined to be the multiset $\{u\}$; the complexity of $u \Leftarrow v$, the multiset $\{v\}$; and the complexity of $u \Leftrightarrow v$, the multiset $\{u, v\}$. The complexity of a chain is the multiset of the complexities of all its individual steps, i.e., it is a multiset of multisets. We compare two chains by comparing their respective complexities in the two-fold multiset extension of the (well-founded) ordering $\succ$ and denote the resulting ordering by $\succ_\pi$. It is a *proof ordering* in that (i) any proper subproof of a proof is a smaller proof and (ii) the replacement of any subproof by a smaller proof will result in a smaller proof. A proof of $u\,S\,v$ in $I$ is said to be *minimal* (with respect to $\succ_\pi$) if there exists no smaller proof of the same type in $I$.

PROPOSITION 2. *Let $\succ$ be a total well-founded ordering on ground terms, $\succ_\pi$ be the corresponding proof ordering, and $I$ be a Herbrand interpretation. If no peak, plateau, or plain in $I$ is a minimal proof, then all minimal proofs in $I$ are rewrite proofs. Furthermore, if a peak, plateau or plain commutes in $I$, then it is non-minimal.*

PROOF. Direct inspection shows that any rewrite proof is simpler (according to $\succ_\pi$) than any peak, plateau, or plain of the same type. If a proof contains a peak, plateau, or plain as a subproof, then that subproof is non-minimal and hence can be replaced by a simpler proof. The result is a simpler proof of the same type, which implies that the original proof is non-minimal. Thus all minimal proofs must be rewrite proofs. $\square$

As a corollary we obtain the following lemma that provides the basis of a refined chaining calculus for sets $\mathsf{Tr_P}$ of transitivity laws. (We will later prove a similar "commutation lemma" for general sets of composition laws $\mathsf{CL_P}$.)

LEMMA 1. *Let $\succ$ be a total, well-founded ordering on ground terms. The rewrite closure of $I$ with respect to a set of transitivity laws $\mathsf{Tr_P}$ is a model of $\mathsf{Tr_P}$ if and only if all peaks in $I$ commute.*

PROOF. It can easily be seen that if the rewrite closure of $I$ is a model of $\mathsf{Tr_P}$, then all peaks in $I$ commute. For the other direction, it is sufficient by Proposition 1 to show that the rewrite closure of $I$ is the same as the transitive closure. Suppose all peaks in $I$ commute. First note that if a proof contains at least two steps, then any one identity step can be deleted, the result being a simpler (and shorter) proof of the same type. This implies that no plateau or plain is minimal. By assumption, peaks commute and hence are also non-minimal. We may use Proposition 2 to infer that all minimal proofs are rewrite proofs. In short, if an atom $u\,S\,v$ is provable in $I$, then it also has a rewrite proof. Consequently, the rewrite closure of $I$ is the same as the transitive closure. $\square$

For example, if $I$ contains the atoms $a\,S\,b$ and $b\,S\,c$ and $c\,S\,d$, and if $a \succ b \succ c \succ d$, then $a \Rightarrow_S b$ and $b \Rightarrow_S c$ and $c \Rightarrow_S d$. That is, there are no right-to-left rewrite steps and thus all peaks commute (there are none). If we choose the ordering $b \succ c \succ d \succ a$, the rewrite steps are $a \Leftarrow_S b$ and $b \Rightarrow_S c$ and $c \Rightarrow_S d$, and there is a peak, $a \Leftarrow_S b \Rightarrow_S c$, that does not commute. But if we add the atom $a\,S\,c$, then the peak commutes (in the extended interpretation) by virtue of the additional rewrite step $a \Leftarrow_S c$. Note that $a\,S\,c$ can be deduced from $a\,S\,b$ and $b\,S\,c$ by a restricted version of chaining in which the middle term is maximal.

As another example, consider the two unit clauses $x\,S\,ffx$ and $ffx\,S\,hx$ and let $\succ$ be an ordering that compares ground expressions according to their size (and expressions of the same size according to some other criterion). We may use chaining to deduce $x\,S\,hx$, so as to make any peak $u \Leftarrow_S ffu \Rightarrow_S hu$, where $u$ is a ground term, commute. In fact, if $R$ denotes the set of all ground atoms obtained from these three unit clauses by instantiation, then the rewrite closure $I$ of $R$ is a transitivity interpretation for $S$.

Other unit clauses, such as $ffx\,S\,ffhx$, can also be deduced by chaining, but none of them is needed for commutation of any peaks. We can prove that a clause is a logical consequence of the above three clauses and the transitivity law by deriving a contradiction from the Skolemized version of the negated clause, e.g., $\neg ffc\,S\,ffhc$ (where $c$ is a Skolem constant). If there is a one-step proof of $ffc\,S\,ffhc$, that is, if the atom is a ground instance of some other atom, then we get a contradiction by resolution with $\neg ffc\,S\,ffhc$. Since this is not the case here, we may check whether there is a rewrite proof of $ffc\,S\,ffhc$. The search for a rewrite proof is accomplished by what we call negative chaining. More specifically, we check whether any instance of a maximal term is reducible. For instance,

$$\frac{\neg(ffc\,S\,ffhc) \quad x\,S\,ffx}{\neg(ffc\,S\,hc)}$$

is a negative chaining inference that encodes an inference with the transitivity law,

$$\frac{\neg(y\,S\,y'),\, \neg(y'\,S\,z),\, y\,S\,z \quad \neg(ffc\,S\,ffhc) \quad x\,S\,ffx}{\neg(ffc\,S\,hc)}$$

with a most general unifier $\sigma$, such that $x\sigma = y'\sigma = hc$, $y\sigma = ffc$, and $z\sigma = ffhc$. A resolution inference with the conclusion of the negative chaining inference and the clause $ffx\,S\,hx$ yields the empty clause (a contradiction).

Ordered and negative chaining will be formally defined below. Informally, we use ordered chaining to deduce enough atoms so that certain peaks commute. In a refutational theorem proving setting, a negative literal $\neg A$ can by repeatedly simplified via negative chaining until eventually a contradiction is obtained by resolution (if $A$ logically follows from the given clauses). This process reflects the search for a rewrite proof of $A$, with the final resolution inference signifying the (successful) completion of the search. We first present a chaining calculus for Horn clauses. The design of a full chaining calculus for arbitrary clauses and composition laws requires a more sophisticated formalization of these basic ideas.

The use of chaining to achieve commutation of peaks is reminiscent of critical pair computations in the well-known Knuth-Bendix completion procedure. The main difference is that completion applies to (symmetric) congruence relations, whereas we deal with arbitrary transitive relations that need not even be symmetric.[4] The key property underlying critical-pair computation is the (local) confluence of a rewrite system. Commutation is the appropriate generalization of this property to non-symmetric relations. There is an important difference between these two properties, in that convergent (i.e., confluent and terminating) rewrite systems define unique normal-form expressions. Thus, if $R$ is a convergent rewrite system, then there exists a rewrite proof of $u\,R\,v$ if and only if $u$ and $v$ have the same normal form. The search for rewrite proofs is therefore don't-care nondeterministic in that no backtracking is required. Commutation is weaker in this regard, and the search for a rewrite proof, as represented by negative chaining, in general requires

[4]Subterm chaining calculi and congruence relations will be discussed in Section 5.

backtracking. For a discussion of the completion procedures that are most relevant to chaining, such as ordered completion or conditional completion, see Bachmair, Dershowitz, and Plaisted [1989], Ganzinger [1991], and Dershowitz [1991].

### 3.3 Ordered chaining

We will use a well-founded ordering $\succ$ on ground expressions (more precisely, terms and literals) to reason about chaining calculi. In the description of inference rules, on the other hand, we will apply the ordering also to non-ground expressions, with the following standard extension in mind: $E \succ E'$ if and only if $E\sigma \succ E'\sigma$, for all ground instances $E\sigma$ and $E'\sigma$. Thus, if $E'\sigma \succ E\sigma$, for some ground instances $E\sigma$ and $E'\sigma$, then $E \not\succeq E'$.[5] We say that a literal $L$ is *maximal with respect to a clause* $C$ if $L' \succ L$, for no literal $L'$ in $C$; and that $L$ is *strictly maximal* with respect to $C$ if $L' \succeq L$, for no $L'$ in $C$.

The *ordered chaining calculus* $\mathcal{B}^{\succ}_{\mathsf{P}}$, or simply $\mathcal{B}^{\succ}$ or $\mathcal{B}$, consists of the following inference rules.[6]

*Ordered Chaining.*

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

where (i) $S$ is an element of $\mathsf{P}$ and $\sigma$ is the most general unifier of $s$ and $t$, (ii) $u\sigma\,S\,s\sigma$ is strictly maximal with respect to $C\sigma$, (iii) $t\sigma\,S\,v\sigma$ is strictly maximal with respect to $D\sigma$, (iv) $u\sigma \not\preceq s\sigma$, and (v) $v\sigma \not\preceq t\sigma$.

*Negative Chaining.*

$$\frac{C\,,\, u\,S\,s \quad D\,,\, \neg(t\,S\,v)}{C\sigma\,,\, D\sigma\,,\, \neg(s\sigma\,S\,v\sigma)}$$

where (i) $S$ is an element of $\mathsf{P}$ and $\sigma$ is the most general unifier of $u$ and $t$, (ii) $u\sigma\,S\,s\sigma$ is strictly maximal with respect to $C\sigma$, (iii) $\neg(t\sigma\,S\,v\sigma)$ is maximal with respect to $D\sigma$, (iv) $s\sigma \not\preceq u\sigma$; and (v) $v\sigma \not\preceq t\sigma$, and

$$\frac{C\,,\, u\,S\,s \quad D\,,\, \neg(t\,S\,v)}{C\sigma\,,\, D\sigma\,,\, \neg(t\sigma\,S\,u\sigma)}$$

where (i) $S$ is an element of $\mathsf{P}$ and $\sigma$ is the most general unifier of $s$ and $v$, (ii) $u\sigma\,S\,s\sigma$ is strictly maximal with respect to $C\sigma$, (iii) $\neg(t\sigma\,S\,v\sigma)$ is maximal with respect to $D\sigma$, (iv) $u\sigma \not\preceq s\sigma$, and (v) $t\sigma \not\prec v\sigma$.[7]

*Ordered Resolution.*

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

[5] This non-ground extension of an ordering need not be decidable, even if the ground ordering is decidable. For practical purposes it may therefore be necessary to employ a decidable approximation of the non-ground ordering when applying the inference rules given below. The use of a weaker ordering is always sound, but the restrictions imposed on an inference may be weaker.

[6] We implicitly assume that different premises of an inference have no variables in common. If necessary, the variables in one premise are renamed. It may also be possible to use different variants of the same clause as premises in one inference.

[7] The case $t\sigma = v\sigma$ needs to be dealt with by only one of the two negative chaining rules.

where (i) $\sigma$ is the most general unifier of $A$ and $B$, (ii) $A\sigma$ is strictly maximal with respect to $C\sigma$, and (iii) $\neg B\sigma$ is maximal in $D\sigma$.

The choice of the ordering $\succ$ is an important mechanism for restricting inference rules and thereby controlling the proof search of a theorem prover.[8] Another high-level primitive for guiding the proof search is the specification of selection strategies for negative literals. Results about selection in equational theorem proving [Bachmair and Ganzinger 1994] can be extended to the framework considered in the present paper, though we have not included them so as to not complicate the presentation technically.

### 3.4 Clause orderings

Orderings on literals play a key role in the definition of chaining calculi. When reasoning about chaining calculi, and proving their refutational completeness, we need orderings on clauses. We have to impose certain restrictions on these orderings, in part to ensure their compatibility with the proof ordering we used to define minimal proofs (and prove resulting commutation lemmas). The following definitions refer to orderings on ground terms and literals.

An ordering $\succ$ is said to be *compatible with composition* if $L \succ L'$ whenever $L$ contains a special predicate symbol $S$ and $L'$ is obtained from $L$ by replacing $S$ by a smaller (in the ordering $<_{\mathsf{P}}$) predicate $T$.

It is called *compatible with reduction of maximal subterms* if $L \succ L'$ whenever $L$ and $L'$ contain the same special predicate symbol $S$, but the maximal subterm of $L'$ is strictly smaller than the maximal subterm of $L$. For example, if $u \succ v$ and $\succ$ is compatible in this sense, then $u\,S\,v \succ v\,S\,v$ and $u\,S\,v \succ \neg(v\,S\,v)$.

An ordering is called *compatible with goal reduction* if (i) $\neg A \succ A$, for all ground atoms $A$, (ii) $\neg A \succ B$ whenever $A$ is an atom $u\,S\,v$ and $B$ is an atom $u'\,S'\,v'$, such that $S$ is a special predicate and $(u', S', v')$ occurs in some valley chain of type $A$,[9] and (iii) $\neg A \succ \neg B$ whenever $A$ is an atom $u\,S\,u$ and $B$ and atom $u\,S\,v$ or $v\,S\,u$, where $S$ is a special predicate and $u \succ v$. If an ordering is compatible in this sense, and one attempts to refute a "goal" $\neg A$ by searching for a rewrite proof of $A$, then the proof search can be restricted to atoms smaller than the goal.

Finally, an ordering is called *admissible* if it is (a) well-founded and total on ground terms and literals, and (b) compatible with (i) composition, (ii) reduction of maximal subterms, and (iii) goal reduction. An ordering on ground clauses is called *admissible* if it is the multiset extension of an admissible ordering on literals. Specific admissible orderings will be described later.

We say that a ground inference is *decreasing* with respect to a clause ordering $\succ$ if its conclusion is smaller than the maximal premise.

LEMMA 2. *If $\succ$ is an admissible ordering, then any ground inference in $\mathcal{B}^{\succ}$ is decreasing with respect to $\succ$.*

PROOF. For ordered chaining and ordered resolution the assertion can be readily established (using compatibility with reduction of maximal subterms and with goal

[8] For a survey on orderings see Dershowitz [1987].
[9] Property (i) is implied by (ii) if $A$ is an atom with a special predicate symbol.

reduction). Consider a ground negative chaining inference

$$\frac{C\,,\,u\,S\,s \quad D\,,\,\neg(u\,S\,v)}{C\,,\,D\,,\,\neg(s\,S\,v)}$$

where $u \succ s$, $u \succ v$, and $u\,S\,s \succ C$[10] (plus other constraints not relevant here). Since $\succ$ is compatible with reduction of maximal subterms, we may infer that $\neg(u\,S\,v) \succ \neg(s\,S\,v)$. In addition, the chain $(u, S, s), (s, S, v)$ is a valley, so that by compatibility of $\succ$ with goal reduction, we get $\neg(u\,S\,v) \succ u\,S\,s$ and therefore $\neg(u\,S\,v) \succ C$. The conclusion of the inference is therefore smaller than the second premise. Similarly, for a ground negative chaining inference

$$\frac{C\,,\,u\,S\,s \quad D\,,\,\neg(t\,S\,s)}{C\,,\,D\,,\,\neg(t\,S\,u)}$$

we have $s \succ u$, $s \succeq t$, and $u\,S\,s \succ C$. Since $\succ$ is admissible, we may infer that $\neg(t\,S\,s) \succ \neg(t\,S\,u)$. The chain $(t, S, u), (u, S, s)$ is a valley, so that by compatibility of $\succ$ with goal reduction, we get $\neg(t\,S\,s) \succ u\,S\,s$ and therefore $\neg(t\,S\,s) \succ C$. Again, the conclusion of the inference is smaller than the second premise. □

### 3.5 Candidate models

A *Horn clause* is a clause that contains at most one positive literal. We next prove that if an ordering $\succ$ is admissible, then the calculus $\mathcal{B}_{\mathsf{P}}^{\succ}$ is refutationally complete for sets of Horn clauses. We adapt proof techniques from Bachmair and Ganzinger [1990] and Bachmair and Ganzinger [1994].

Given a set $N$ of ground clauses, we define a corresponding Herbrand interpretation $I$ (a "candidate model" for $N$) by induction on $\succ$.[11] More specifically, we define for each clause $C$ an interpretation $I_C$, intended to be a model for clauses (strictly) smaller than $C$, and a set $E_C$ that is designed to turn $I_C$ into a model of $C$. The Herbrand interpretations $I_C$ and $I$ will be represented indirectly, as rewrite closures of sets $R_C$ and $R$, respectively.

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 $I_C$ the rewrite closure of $R_C$. Furthermore, if $C$ is a clause $C' \vee A$, where $A$ is a positive literal with $A \succ C'$, and $C$ is false in $I_C$ then $E_C = \{A\}$. In that case, we also say that $C$ is *productive* (and *produces* $A$). In all other cases, $E_C = \emptyset$. Finally, let $R$ be the set $\bigcup_C E_C$ and $I$ the rewrite closure of $R$. We also use $R^C$ to denote the set $R_C \cup E_C$ and $I^C$ to denote the rewrite closure of $R^C$. From these definitions we immediately obtain:

Lemma 3. *If $C$ is a productive clause in $N$, then it is true in $I^C$.*

An important monotonicity property of our construction is expressed in the following lemma.

Lemma 4. *Let $\succ$ be an admissible ordering. 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 it is also true in $I$ and in all interpretations $I_{D'}$ and $I^{D'}$, where $D' \succ D$ (and $D$ and $D'$ are clauses in $N$).*

[10] When we write $A \succ C'$ we view $A$ as denoting a positive unit clause.

[11] We define candidate models for sets of general clauses, even though in this section we prove refutational completeness for Horn clauses only.

PROOF. Let $C$, $D$ and $D'$ be ground clauses, such that $D' \succ D \succeq C$ and $D$ and $D'$ are elements of $N$. From the above definitions it can easily be seen that $I_D \subseteq I^D \subseteq I_{D'} \subseteq I^{D'} \subseteq I$. Thus, if a positive literal $A$ in $C$ is true in $I_D$ or $I^D$, then $A$ (and hence $C$) is also true in $I_{D'}$, $I^{D'}$ and $I$. If, on the other hand, a negative literal $\neg A$ in $C$ is true in $I_D$ or $I^D$, then $A$ is false in $I_D$ or $I^D$. We claim that $A$ is also false in $I$. Since the clause ordering is admissible, we know that if $B$ is an atom produced by a clause greater than or equal to $D$, then $B \succ \neg A$. Since $\succ$ is compatible with goal reduction, we have $\neg A \succ A$ and also $\neg A \succ A'$, for any atom $A' = u\,S\,v$ for which $(u, S, v)$ may occur in a rewrite proof of type $A$. In other words, no atom $B$ produced by any clause greater than $D$ can possibly be used in a rewrite proof of $A$. This implies that $A$ is false, and $\neg A$ true, in $I$. We conclude that $C$ is true in $I_{D'}$ and $I^{D'}$, as well as in $I$. □

We will often use the contrapositive of this lemma, to infer that $C$ is false in the interpretations $I_C$ and $I^C$ whenever it is false in $I_{D'}$ or $I^{D'}$, for some $D' \succ C$. The next lemma states a continuity property of the model construction.

LEMMA 5. *Let $D$ be a clause in $N$ and $C$ be a ground clause (which need not be in $N$). If $C$ is true in all interpretations $I^{D'}$, where $D \succ D'$, then it is also true in $I_D$.*

PROOF. A clause $C$ is certainly true in $I_D$ if it contains a literal $\neg A$, such that $A$ is false in $I_D$. Suppose, on the other hand, that for all negative literals $\neg A$ in $C$, the atom $A$ is true (and $\neg A$ false) in $I_D$. If an atom $A$ is true in $I_D$, it is either an element of $R_D$ or else has a rewrite proof in $R_D$. This implies that $A$ is true in some interpretation $I^{D_A}$, where $D_A$ is a clause with $D \succ D_A$. Let $C'$ be the maximal of all such clauses $D_A$, where $\neg A$ is a literal in $C$. We know that $C$ is true in $I^{C'}$, since $D \succ C'$. But since all negative literals of $C$ are false in $I^{C'}$, some positive literal of $C$ must be true in $I_{C'}$ and, hence, in $I_D$. We conclude that $C$ is true in $I_D$. □

### 3.6 Refutational completeness for Horn clauses

We say that a set of (possibly non-ground) clauses $N$ is *saturated* (with respect to an inference system $\mathcal{J}$) if it contains all conclusions of inferences (in $\mathcal{J}$) from $N$.

LEMMA 6. *Let $\succ$ be an admissible ordering and $N$ be a set of ground Horn clauses that is saturated with respect to $\mathcal{B}^{\succ}_{\mathsf{P}}$ and does not contain the empty clause. If $I$ is the interpretation constructed from $N$ (with respect to $\mathsf{Tr}_{\mathsf{P}}$), 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) Both $I^C$ and $I_C$ are transitivity interpretations.*
*(3) The clause $C$ is true in $I^C$.*

PROOF. The proof is by induction on $\succ$. Let $C$ be a ground clause in $N$, such that assertions (1)–(3) are satisfied for all clauses in $N$ that are smaller than $C$.

(1) If $C$ is a productive clause $C' \vee A$, where $A \succ C'$, then it it is false in $I_C$ and $C'$ contains only negative literals, $C$ being a Horn clause. Therefore, for every literal $\neg B$ in $C'$, we know that $B$ is true in $I_C$ and, by Lemma 4, true in $I$. This implies that $C'$ is false in $I$.

(2) By Lemma 1 it suffices to prove that all peaks in $R_C$ and $R^C$, respectively, commute. Each peak in $R_C$ is a peak in $R^{C'}$, for some $C'$ with $C \succ C'$, and commutes in $R^{C'}$ by the induction hypothesis. Thus it also commutes in $R_C$. Let us consider peaks in $R^C$ that are not peaks in $R_C$. If $E_C$ is empty, there are no such peaks. Let us therefore assume that $C$ is productive. Let $u \Leftarrow_S t \Rightarrow_S v$ be a peak in $R^C$. Then there exist a clause $C_1 \vee u\,S\,t$ that produces $u\,S\,t$, and another clause $C_2 \vee t\,S\,v$ that produces $t\,S\,v$. (The two clauses are not identical and the larger of the two is $C$.) From these two clauses we obtain $C' = C_1 \vee C_2 \vee u\,S\,v$ by ordered chaining. Since $N$ is saturated, $C'$ is a clause in $N$. Since the ordering $\succ$ is admissible, we may use Lemma 2 to infer that $C \succ C'$. Thus, by part (3) of the induction hypothesis, $C'$ is true in $I^{C'}$. By part (1), both $C_1$ and $C_2$ are false in $I$ and, hence, in $I^{C'}$. Therefore $u\,S\,v$ must be true in $I_{C'}$, which implies $u \Downarrow_S^{R_{C'}} v$ and $u \Downarrow_S^{R_C} v$. That is, the peak commutes in $R_C$.

(3) If $C$ is false in $I^C$, it is non-productive and hence must violate one of the conditions imposed on productive clauses. The only condition that can be violated is that the maximal literal in $C$ is not positive. In other words, $C$ is a clause $C' \vee \neg A$, where $\neg A$ is a maximal literal and $A$ 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 $N$ (since $N$ is saturated), and false in $I_C$ (since by part (1), $D'$ is false in $I$). But then $C''$ is also false in $I^{C''}$, which contradicts part (3) of the induction hypothesis. The only remaining possibility is that $A$ is an atom $u\,S\,v$ that has a rewrite proof with at least two steps in $R_C$, i.e., $u \Downarrow_S^{R_C} v$. If $u \succ v$, then there exists a productive clause $D \vee u\,S\,v'$, such that $v' \Downarrow_S^{R_C} v$; and if $v \succeq u$, there exists a productive clause $D \vee u'\,S\,v$, such that $u \Downarrow_S^{R_C} u'$. In either case we may obtain, by negative chaining with $C$, a clause $C''$ smaller than $C$ that is contained in $N$ and false in $I^{C''}$, which is again a contradiction. In sum, we may conclude that $C$ is true in $I^C$, which completes the proof. □

As an immediate corollary of the lemma we obtain:

THEOREM 1. *If a set of Horn clauses $N$ is saturated with respect to $\mathcal{B}_{\mathsf{P}}^{\succ}$ and does not contain the empty clause, then $N \cup \mathsf{Tr}_{\mathsf{P}}$ is satisfiable.*

PROOF. Suppose $N$ does not contain the empty clause. We claim that the Herbrand interpretation $I$ constructed from the set of all ground instances of $N$ is a model of $N \cup \mathsf{Tr}_{\mathsf{P}}$. Standard results in resolution theory (usually called "lifting lemmas") imply that if a set of clauses $N$ is saturated with respect to $\mathcal{B}_{\mathsf{P}}^{\succ}$, then the set of all its ground instances is also saturated. By the above lemmas, every ground instance $C$ of a clause in $N$ is true in $I^C$ and in $I$, and in addition $I$ is a transitivity interpretation. □

This theorem suggests saturation as a basis for checking the unsatisfiability of sets of Horn clauses over languages with transitive relations. Saturate a given set of clauses with respect to the ordered chaining calculus (by exhaustively applying all possible inferences): if the initial clause set, along with corresponding transitivity laws, is unsatisfiable, a contradiction in the form of the empty clause will eventually be generated. In the following sections we will generalize ordered chaining to non-Horn clauses and to composition laws. First, though, we briefly discuss the issue of variable chaining.

## 3.7 Variable chaining

Ordered and negative chaining inferences use unification as a filter in that an argument of a special predicate in the first premise has to be unified with a suitable argument of special predicate in the second premise. If one of the two terms thus unified is a variable we speak of a *variable chaining* (or chaining *through* a variable). More specifically, we speak of chaining *into* a variable if the literal with the variable is maximal (among the two literals), and of chaining *from* a variable otherwise. Since a term can always be unified with a variable not occurring in it, variable chaining can be extremely prolific. But with the right combination of redundancy criteria (to be defined in the next section) and ordering constraints one may drastically cut down the number of possible variable chaining inferences.

Consider, for instance, a chaining inference

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

where $x\sigma = t$. Ordered chaining imposes various constraints on the terms and literals in the premises. These constraints imply that $x$ is a maximal term in the first premise. If one defines the ordering in a suitable way, one obtains that $x$ may occur neither in a negative literal nor as a nested subterm, say in a term $f(\ldots x \ldots)$. We also say that $x$ is *unshielded* in that case.

The inference can also be shown to be redundant if the variable $x$ is *linear* (i.e., occurs only once) in the first premise, for in that case the conclusion is $C \vee D \vee u\,S\,v$, and is either properly subsumed by $C \vee u\,S\,x$ or else is identical to $C \vee u\,S\,x$ up to renaming of variables. Similar arguments apply to inferences

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

where the variable occurs as the first argument of a literal in the second premise.

In short, ordered chaining through a variable is only necessary if the variable is non-linear, unshielded, and occurs only in positive literals. One can easily check that in the case of Horn clauses, the combination of these conditions excludes all non-trivial ordered chaining inferences through variables. 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

$$\begin{aligned} &\to\ a\,S\,a' \\ &\to\ a'\,S\,c \\ &\to\ b\,S\,b' \\ &\to\ b'\,S\,c \\ a\,S\,x\,,\,b\,S\,x\ &\to \end{aligned}$$

where $c \succ b' \succ b \succ a' \succ a$. This set of clauses is unsatisfiable. From the first two clauses we obtain $a\,S\,c$; from the third and fourth clause $b\,S\,c$. These two unit clauses contradict the last clause. No ordered chaining is possible. We may resolve the third with the last clause, to get $\neg(a\,S\,b')$, from which we get $\neg(a\,S\,b)$ by chaining with the third clause again. All 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 $\neg(a\,S\,c) \vee \neg(b\,S\,b')$, from which a contradiction can be derived by another chaining and two resolution steps.

In Sections 5.3 and 6.3 we show that chaining into variables can be avoided for congruence relations and for dense, total orderings without endpoints.

## 4. COMPOSITE BINARY RELATIONS

We generalize the results of the previous section in several directions by considering non-Horn clauses and general composition laws for binary relations, and formulating a suitable concept of redundancy. Redundancy will also allow us to exploit additional logical properties of relations when designing specialized derived inferences to render certain generic inferences redundant.

The following commutation lemma provides the basis for the general ordered chaining calculus.

LEMMA 7. *The rewrite closure of $I$ with respect to a set of composition laws $\mathsf{CL_P}$ is a model of $\mathsf{CL_P}$ if and only if all peaks and plateaus in $I$ commute.*

PROOF. It can easily be proved that if the rewrite closure of $I$ is a model of $\mathsf{CL_P}$, then all peaks and plateaus in $I$ commute. For the other direction, we show that the rewrite closure of $I$ is the same as the composition closure. First note that if a proof contains two successive identity steps, then the step with the smaller (according to $<_{\mathsf{P}}$) predicate symbol can be deleted to yield a simpler (and shorter) proof of the same type. This implies that no plain is minimal. By assumption, peaks and plateaus commute and hence are also non-minimal. By Proposition 2, all minimal proofs are therefore rewrite proofs. We conclude that the rewrite closure of $I$ is a model of $\mathsf{CL_P}$. $\square$

### 4.1 Redundancy

We present an abstract notion of redundancy that derives from our earlier work [Bachmair and Ganzinger 1994]. Let $\mathsf{T}$ be a set of clauses (e.g., a collection of composition laws) intended to be built into an inference system. Intuitively, by redundancy with respect to a fixed set of clauses $\mathsf{T}$ we want to characterize, given any set of clauses $N$, when the deletion of a clause will not prevent a refutational theorem prover from deriving a contradiction from the set $N$, or when an inference need not be applied to obtain a refutation. The notion of redundancy we propose exploits properties of candidate models and the refutational completeness proofs based thereon, and therefore depends on a given clause ordering.

Observe, for instance, that in the proof of Lemma 6 all essential inferences involve only ground clauses $C$ that are false in the candidate model $I_C$ and for which smaller clauses are already known to be true in $I_C$. If a ground clause cannot possibly satisfy these properties for any candidate model $I$, then it is redundant. Unfortunately, a characterization of redundancy in terms of candidate models usually allows for no effective methods for testing redundancy. We will therefore settle for a weaker notion of redundancy that does not explicitly refer to candidate models, but nevertheless is general enough to cover virtually all redundancy criteria that have been proposed for saturation-based theorem provers.

We say that a ground clause $C$ is $\mathsf{T}$-*redundant* with respect to $N$ if there exists a set $\{C_1, \ldots, C_k\}$ of ground instances of $N$ such that $C$ is true in every $\mathsf{T}$-model of $\{C_1, \ldots, C_k\}$ and $C \succ C_j$, for all $j$ with $1 \leq j \leq k$. A non-ground clause is called $\mathsf{T}$-*redundant* if all its ground instances are.

Note that a clause $C$ may be redundant with respect to $N$ even if it is not an element of $N$. In fact, tautologies are $\mathsf{T}$-redundant with respect to any set $N$. By definition any element of $\mathsf{T}$ is $\mathsf{T}$-redundant. Proper subsumption is also covered by this notion of redundancy: a clause $C\sigma \vee D$ is $\mathsf{T}$-redundant with respect to any set $N$ containing the clause $C$, provided $D$ is not the empty clause.

We call the maximal premise of a (ground) inference the *main premise*, and any smaller premise, a *side premise*. (For the inference rules we discuss, there is at most one side premise). The maximal literal of the main premise of an inference is called its *main literal.* The *side literal* of an inference is defined as (i) the maximal literal of the side premise, if there is a side premise, and (ii) the constant $\top$ (which is defined to be true in all interpretations), otherwise. For instance, if an inference has only one premise or has two identical premises (see composition resolution below), then its side literal is $\top$. Further inspection of the proof of Lemma 6 reveals that the side premise of any essential two-premise ground inference is a productive clause $C$, whereas the main premise is usually a counterexample to the candidate model.

A ground inference with conclusion $B$, main premise $C$, and side literal $L$ is said to be $\mathsf{T}$-*redundant* with respect to $N$ if either one of its premises is redundant, or else there exists a set $\{C_1, \ldots, C_k\}$ of ground instances of $N$ such that $B$ is true in every $\mathsf{T}$-model of $\{L, C_1, \ldots, C_k\}$ and $C \succ C_j$, for all $j$ with $1 \leq j \leq k$. A non-ground inference is called $\mathsf{T}$-redundant if all of those ground instances that satisfy the various side conditions of the inference are redundant. We say that a set of clauses $N$ is *saturated up to* $\mathsf{T}$-*redundancy* with respect to some inference system, if all inferences from $N$ are $\mathsf{T}$-redundant.

This notion of redundancy is compatible with chaining calculi in several ways. For one thing, redundancy (of clauses and inferences) is preserved under deduction of new clauses and deletion of redundant clauses. That is, if a clause or inference is redundant with respect to a set $N$, it is also redundant with respect to any set $N \setminus \{C\}$, where $C$ is redundant with respect to $N$, and any set $N \cup \{D\}$. In addition, any set of clauses $N$ that is saturated up to redundancy is unsatisfiable if and only if it contains the empty clause. A decreasing inference can be rendered redundant by applying it, i.e., adding its conclusion to the set of deduced clauses. Thus, if all ground inferences are decreasing, then for any recursively enumerable set of clauses an equivalent, saturated set can be recursively enumerated. The question of how to saturate a clause set up to redundancy is independent of the specifics of chaining calculi, and we refer the reader to Bachmair and Ganzinger [1994] and Bachmair and Ganzinger [1997] for a detailed exposition of the relevant issues, such as fairness, simplification techniques, and redundancy detection.

### 4.2 Ordered chaining for general composition laws

Let $\mathsf{P}$ be a set of special predicate symbols and $\succ$ be an ordering on ground expressions. The general *ordered chaining calculus* $\mathcal{C}^{\succ}_{\mathsf{P}}$ (or $\mathcal{C}^{\succ}$ or simply $\mathcal{C}$) is an extension of the calculus $\mathcal{B}^{\succ}_{\mathsf{P}}$ by the following inference rules:

*Ordered Chaining.*

$$\frac{C\,,\,u\,S\,s \quad D\,,\,t\,T\,v}{C\sigma\,,\,D\sigma\,,\,u\sigma\,U\,v\sigma}$$

where (i) $S$ and $T$ are comparable and $U$ is the maximum of $S$ and $T$, (ii) $\sigma$ is the most general unifier of $s$ and $t$; (iii) $u\sigma\,S\,s\sigma$ is strictly maximal with respect to $C\sigma$; (iv) $t\sigma\,T\,v\sigma$ is strictly maximal with respect to $D\sigma$; (v) if $U = T$, then $u\sigma \not\preceq s\sigma$ and $v\sigma \not\succ t\sigma$; and (vi) if $U = S$, then $u\sigma \not\succ s\sigma$ and $v\sigma \not\preceq t\sigma$.

*Negative Chaining.*

$$\frac{C\,,\,u\,S\,s \quad D\,,\,\neg(t\,T\,v)}{C\sigma\,,\,D\sigma\,,\,\neg(s\sigma\,U\,v\sigma)}$$

where (i) $S \le_{\mathsf{P}} T$ and $U \le_{\mathsf{P}} T$, (ii) $\sigma$ is the most general unifier of $u$ and $t$; (iii) $u\sigma\,S\,s\sigma$ is strictly maximal with respect to $C\sigma$; (iv) $\neg(t\sigma\,T\,v\sigma)$ is maximal with respect to $D\sigma$; (v) $s\sigma \not\preceq u\sigma$, and (vi) $v\sigma \not\preceq t\sigma$; and

$$\frac{C\,,\,u\,S\,s \quad D\,,\,\neg(t\,T\,v)}{C\sigma\,,\,D\sigma\,,\,\neg(t\sigma\,U\,u\sigma)}$$

where (i) $S \le_{\mathsf{P}} T$ and $U \le_{\mathsf{P}} T$, (ii) $\sigma$ is the most general unifier of $s$ and $v$; (iii) $u\sigma\,S\,s\sigma$ is strictly maximal with respect to $C\sigma$; (iv) $\neg(t\sigma\,T\,v\sigma)$ is maximal with respect to $D\sigma$; (v) $u\sigma \not\preceq s\sigma$, and (vi) $t\sigma \not\succ v\sigma$.

For general clauses we also need a factoring rule:

*Ordered (Positive) Factoring.*

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

where (i) $\sigma$ is the most general unifier of the atoms $A$ and $B$ and (ii) $A\sigma$ is maximal with respect to $C\sigma$.

The soundness of the above inference rules can be checked easily. The inference rules are unfortunately insufficient for disjunctions of positive literals with multiple occurrences of the maximal term of a clause. For example, the set of clauses

$$\begin{array}{lrl}
(1) & & \to\; c \le b \\
(2) & & \to\; b \le c \\
(3) & & \to\; a \le b,\, a \le c \\
(4) & a \le b,\, a \le c & \to
\end{array}$$

where $\le$ is transitive and $a \succ b \succ c$, is unsatisfiable. The third clause requires that $a \le b$ or $a \le c$ be true. But by the first two clauses, if one of the two inequalities is true, so is the other, which contradicts the last clause. By ordered chaining we obtain

$$(5) \qquad\qquad \to\; c \le c$$

from the first two clauses. From clauses (3) and (4) one can only obtain a tautology, which is redundant. Thus, the set of clauses (1)–(5) is saturated up to redundancy with respect to the above inference rules, but does not contain the empty clause. The following inference rules restore refutational completeness.

*Composition Resolution.*

$$\frac{C\,,\,u\,S\,v\,,\,u'\,S\,v' \quad D\,,\,s\,S\,t}{D\sigma\,,\,\neg(t\sigma\,T\,v\sigma)\,,\,s\sigma\,S\,v\sigma}$$

where (i) $T \leq_{\mathsf{P}} S$; (ii) $\sigma$ is the most general unifier of $u$, $u'$ and $s$; (iii) $u'\sigma\,S\,v'\sigma$ is strictly maximal with respect to $C\sigma \vee u\sigma\,S\,v\sigma$; (iv) the literal $s\sigma\,S\,t\sigma$ is strictly maximal with respect to $D\sigma$; (v) $t\sigma \not\succeq s\sigma$; and (vi) $u'\sigma\,S\,v'\sigma \not\succ s\sigma\,S\,t\sigma$, and

$$\frac{C\,,\,u\,S\,v\,,\,u'\,S\,v' \quad D\,,\,s\,S\,t}{D\sigma\,,\,\neg(u\sigma\,T\,s\sigma)\,,\,u\sigma\,S\,t\sigma}$$

where (i) $T \leq_{\mathsf{P}} S$; (ii) $\sigma$ is the most general unifier of $v$, $v'$ and $t$; (iii) $u'\sigma\,S\,v'\sigma$ is strictly maximal with respect to $C\sigma \vee u\sigma\,S\,v\sigma$; (iv) the literal $s\sigma\,S\,t\sigma$ is strictly maximal with respect to $D\sigma$; (v) $s\sigma \not\succeq t\sigma$; and (vi) $u'\sigma\,S\,v'\sigma \not\succ s\sigma\,S\,t\sigma$.

Continuing the above example, we find that the third clause, by composition resolution with itself, yields

$$(6) \qquad b \leq c \;\rightarrow\; a \leq c.$$

Now we obtain a refutation as follows:

| | | |
|---|---|---|
| (7) | $b \leq c,\, c \leq b,\, a \leq c \;\rightarrow$ | [negative chaining of (6) into (4)] |
| (8) | $b \leq c,\, b \leq c,\, c \leq b,\, c \leq c \;\rightarrow$ | [negative chaining of (6) into (7)] |
| (9) | $c \leq c \;\rightarrow$ | [repeated resolution with (1) and (2) from (8)] |
| (10) | $\rightarrow$ | [resolution of (5) and (9)] |

Composition resolution is a sound inference rule that represents a controlled application of resolution with the composition laws. For instance, consider the resolution inference,

$$\frac{D\sigma\,,\,s\sigma\,S\,t\sigma \quad \neg(x\,S\,y)\,,\,\neg(y\,T\,z)\,,\,xSz}{D\sigma\,,\,\neg(t\sigma\,T\,z)\,,\,s\sigma\,S\,z.}$$

The conclusion of the first variant of composition resolution is an instance of the conclusion of this inference. The first premise of a composition resolution inference regulates which instances and resolvents are needed. Composition resolution is not a very appealing inference rule, but as we shall see can be drastically curtailed in the presence of additional logical properties, such as symmetry or totality.

### 4.3 Refutational completeness

Not all the inferences of the ordered chaining calculus are decreasing with respect to arbitrary admissible orderings. We impose additional conditions on orderings via a complexity measure on literals $u\,S\,v$ or negations thereof, where $S$ is a special predicate. We denote by $max_L$ the term $u$, if $u \succeq v$, and the term $v$, otherwise. By $s_L$ we denote the predicate symbol $S$ of $L$. We also let $p_L$ be 1, if the literal $L$ is negative, and 0 otherwise; let $a_L$ be 0 if $v \succ u$, and 1 otherwise; and let $d_L$ be 1 if $u = v$, and 0 otherwise. An admissible ordering $\succ$ is said to be *strongly admissible* (for a set of special predicates $\mathsf{P}$) if $L \succ L'$ whenever $L$ and $L'$ contain predicates from $\mathsf{P}$ and the quintuple $(max_{L'}, p_{L'}, s_{L'}, a_{L'}, d_{L'})$ associated with $L'$ is strictly smaller than the quintuple $(max_L, p_L, s_L, a_L, d_L)$ associated with $L$. (Tuples are compared lexicographically, using the ordering $\succ$ in the first component, the ordering $<_{\mathsf{P}}$ on predicate symbols in the third component, and the ordering $1 > 0$ in the other components.)

LEMMA 8. *The inference rules of the ordered chaining calculus $\mathcal{C}^{\succ}$ satisfy the following properties:*

*(1) All ground negative chaining and ground ordered factoring inferences are decreasing if $\succ$ is admissible.*

*(2) A ground ordered chaining inference in which the main and side literals contain different special predicates $S$ and $T$ is decreasing if $\succ$ is strongly admissible for a set of symbols containing $S$ and $T$.*[12]

*(3) A ground composition resolution inference in which the main literal contains the special predicate $S$ is decreasing if $\succ$ is strongly admissible for some set of symbols containing $S$.*

PROOF. (1) For ordered factoring the proof is straightforward; for negative chaining one may use similar arguments as in the proof of Lemma 2.

(2) Consider a ground instance of ordered chaining,

$$\frac{C, u\,S\,s \quad D, s\,T\,v}{C, D, u\,U\,v,}$$

where (i) $U$ is the maximum of $S$ and $T$ and either $S <_{\mathsf{P}} T$ or $T <_{\mathsf{P}} S$, (ii) $u\,S\,s \succ C$, (iii) $s\,T\,v \succ D$, (iv) $s \succeq u$ and $s \succ v$ if $T <_{\mathsf{P}} S$, and (v) $s \succ u$ and $s \succeq v$ if $S <_{\mathsf{P}} T$. We distinguish several subcases.

If $s \succ u$ and $s \succ v$, we may use similar arguments as in the proof of Lemma 2.

If $s = u$, then $s \succ v$ and $T <_{\mathsf{P}} S$. In that case we may use strong admissibility to infer that $u\,S\,s \succ s\,T\,v$ and $u\,S\,s \succ u\,U\,v$. The conclusion is thus smaller than the first premise.

If $s = v$, then $s \succ u$ and $S <_{\mathsf{P}} T$. We have $s\,T\,v \succ u\,S\,s$ and $s\,T\,v \succ u\,U\,v$, so that the conclusion is smaller than the second premise.

(3) Let us consider a ground instance of the first variant of composition resolution,

$$\frac{C, s\,S\,v, s\,S\,v' \quad D, s\,S\,t}{D, \neg(t\,T\,v), s\,S\,v,}$$

where (i) $T \leq_{\mathsf{P}} S$, (ii) $s\,S\,t \succeq s\,S\,v' \succ s\,S\,v$, and (iii) $s \succ t$ and $s \succeq v$. Since $s$ is the maximal term in both premises and the ordering $\succ$ is strongly admissible, we may infer that $s \succ v$ (for $s = v$ would imply $s\,S\,v \succeq s\,S\,v'$). Therefore we have $s\,S\,t \succ \neg(t\,S\,v) \succeq \neg(t\,T\,v)$. The conclusion of the inference is thus smaller than the second premise. Similar arguments apply to the second variant of composition resolution. $\square$

The following key lemma requires slightly weaker conditions than the strong admissibility of the given ordering, which will be useful for further optimizations in Section 6. We say that an admissible ordering $\succ$ is *left-to-right admissible* (for a set of predicate symbols $\mathsf{P}$) if $L \succ L'$ whenever $L$ and $L'$ contain symbols from $\mathsf{P}$ and the quadruple $(max_L, p_L, s_L, a_L)$ associated with $L$ is larger than the quadruple

[12] By Lemma 2 an ordered chaining inference in which main and side literal contain the same predicate symbol is decreasing whenever $\succ$ is admissible.

$(max_{L'}, p_{L'}, s_{L'}, a_{L'})$ associated with $L'$. The lemma refers to candidate models as defined in Section 3.5.

LEMMA 9. *Let $\succ$ be a left-to-right admissible ordering and $N$ be a set of clauses that is saturated up to redundancy with respect to $\mathcal{C}_{\mathsf{P}}^{\succ}$, but does not contain the empty clause. Furthermore, let $I$ be the interpretation constructed from the set of 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 (with respect to $\mathsf{CL_P}$).*

*(2) If a clause $D = D' \vee A$ produces $A$ and $C \succeq D$, then $D'$ is false in $I^C$.*

*(3) The interpretations $I_C$ and $I^C$ are models of $\mathsf{CL_P}$.*

*(4) The clause $C$ is true in $I^C$.*

PROOF. The proof is by induction on $\succ$. Let $C$ be a ground instance of $N$ such that assertions (1)–(4) are satisfied for all smaller ground instances of $N$.

(1) We prove the contrapositive statement. Suppose $C$ is redundant in $N$; that is, there exist smaller ground instances $C_1, \ldots, C_n$ of $N$ such that $C$ is true in every $\mathsf{CL_P}$-model of $\{C_1, \ldots, C_n\}$. We may use parts (3) and (4) of the induction hypothesis and Lemma 4 to infer that $I_C$ is a $\mathsf{CL_P}$-model of $\{C_1, \ldots, C_n\}$. Therefore $C$ is true in $I_C$, which implies that it is non-productive.

(2) Let $D$ be a productive clause $D' \vee A$, where $A \succ D'$ and $C \succeq D$. If $C \succ D$, we may use the induction hypothesis to infer that $D'$ is false in $I^{C'}$, for all ground instances $C'$ of $N$ with $C \succ C' \succeq D$. Consequently $D'$ is false in $I_C$ in that case. If $C = D$, then $D'$ is also false in $I_C$ (since $D$ is productive). Suppose $D'$ is true in $I^C$. Then some literal $L'$ of $D'$ must be true in $I^C$, but false in $I_C$. This is only possible if $C$ is productive and $L'$ is a positive literal $A'$. Suppose $C$ produces $B$, so that $B \succeq A \succ A'$, and $B$ can be used in a rewrite proof of $A'$ (with respect to $\mathsf{CL_P}$). This means that (i) $A'$ is an atom $u\,S\,v$ and (ii) there exist predicate symbols $T$ and $U$, such that $S$ is the maximum of $T$ and $U$, and $B$ is either an atom $u\,U\,u'$, where $u \succ u'$ and $u' \Downarrow_T^{R_C} v$, or an atom $v'\,Uv$, where $v \succ v'$ and $u \Downarrow_T^{R_C} v'$. Let us consider the case where $B$ is of the form $u\,U\,u'$ and $u \succ u'$ and $u' \Downarrow_T^{R_C} v$; the other case is similar.

Since $B \succeq A \succ A'$ and the ordering $\succ$ is left-to-right admissible, we may infer that $U \not<_{\mathsf{P}} S$. In other words, we have $S = U$, so that $A$ must be an atom $u\,S\,w$. But then there is an inference by composition resolution,

$$\frac{D'', u\,S\,v, u\,S\,w \quad C', u\,S\,u'}{C', \neg(u'\,T\,v), u\,S\,v}$$

where $U \leq_{\mathsf{P}} S$. The side literal $L$ of this inference is $A$, if the two premises $C$ and $D$ are different, and $\top$, otherwise. Both $C$ and $D$ are productive and, by part (1), non-redundant. Since $N$ is saturated up to redundancy, there exist clauses $C_1, \ldots, C_n$ smaller than $C$, such that the conclusion $\hat{C}$ of the above inference is true in every $\mathsf{CL_P}$-model of $\{L, C_1, \ldots, C_n\}$. We may use Lemma 4 and parts (3) and (4) of the induction hypothesis to infer that $I_C$ is one such model. Thus the clause $\hat{C}$ is true in $I_C$, but since $C$ is productive, the subclause $C'$ is false in $I_C$. Moreover, we have $u' \Downarrow_T^{R_C} v$, which implies that $\neg(u'\,T\,v)$ is false in $I_C$. Consequently, $u\,S\,v$ must be true in $I_C$, which contradicts the fact that $D'$ is false in $I_C$. We conclude that $D'$ is false in $I^C$.

(3) By Lemma 7 it suffices to show that all peaks and plateaus in $R_C$ and $R^C$, respectively, commute. Each peak or plateau in $R_C$ is a proof in $R^{C'}$, for some $C'$ with $C \succ C'$, and hence commutes in $R^{C'}$ by the induction hypothesis. Thus it also commutes in $R_C$. If $E_C$ is non-empty, then there may be peaks or plateaus in $R^C$, which are not provable in $R_C$. In that case $C$ is productive. Let $u\,S\,t, t\,T\,v$ be a peak or plateau in $R^C$. Then there exist a clause $C_1 \vee u\,S\,t$ that produces $u\,S\,t$, and a (different) clause $C_2 \vee t\,S\,v$ that produces $t\,S\,v$. Both clauses are non-redundant by part (1) of the induction hypothesis, and $C$ is the larger of the two clauses. From these two clauses we obtain $C' = C_1 \vee C_2 \vee u\,U\,v$ by ordered chaining, where $U$ is the maximum of $S$ and $T$. Let $L$ be the side literal of the inference. Since $N$ is saturated up to redundancy, there exist clauses $C_1, \ldots, C_n$ smaller than $C$, such that $C''$ is true in every $\mathsf{CL_P}$-model of $\{L, C_1, \ldots, C_n\}$. We may use Lemma 4 and parts (3) and (4) of the induction hypothesis to infer that $I_C$ is one such model. Thus the clause $C''$ is true in $I_C$, but since the above two clauses are productive, by part 2 the two subclauses $C'$ and $D'$ are false in $I_C$. Therefore the atom $u\,U\,v$ must be true in $I_C$, that is, $u \Downarrow_U^{R_C} v$, which indicates that the peak or plateau commutes.

(4) 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 model of $\mathsf{CL_P}$. Therefore, if $C$ is redundant, it is true in $I^C$. If $C$ is productive, it is true in $I^C$ by definition. Suppose $C$ is neither redundant nor productive. We distinguish two subcases.

(4.1) First suppose the maximal literal in $C$ is negative, so that $C$ can be written as $C' \vee \neg A$, where $\neg A$ is a maximal literal and $A$ is true in $I_C$.

If $A$ is produced by some clause $D = D' \vee A$, then the resolvent $C'' = D' \vee C'$ of $C$ and $D$ is smaller than $C$ (because $\neg A \succ D'$) and $A$ is the side literal of the inference. Since $N$ is saturated up to redundancy, there exist clauses $C_1, \ldots, C_n$ smaller than $C$, such that $C''$ is true in every $\mathsf{CL_P}$-model of $\{A, C_1, \ldots, C_n\}$. We may use the induction hypothesis to infer that $I_C$ is one such model. By part (2), the subclause $D'$ is false in $I^C$, which implies that $C'$, and hence $C$, is true in $I^C$.

If $A$ is an atom $u\,T\,v$ with a rewrite proof of at least two steps in $R_C$, then there exists a productive clause $D' \vee u\,S\,u'$ (where $u \succ u'$ and $u' \Downarrow_U^{R_C} v$) or $D' \vee v'\,S\,v$ (where $v \succ v'$ and $u \Downarrow_U^{R_C} v'$) such that $S \leq_{\mathsf{P}} T$. By negative chaining we get either $C' \vee D' \vee \neg(u'\,U\,v)$ or $C' \vee D' \vee \neg(u\,U\,v')$, where $T$ is the maximum of $S$ and $U$. In either case we may use saturation up to redundancy to infer that $C'$, and hence $C$, is true in $I^C$.

(4.2) If the maximal literal in $C$ is positive, it must have multiple occurrences, for otherwise $C$ would be productive. That is, $C$ can be written as $C' \vee A \vee A$, where $A$ is the maximal literal. The smaller clause $C' \vee A$ can be obtained from $C$ by ordered factoring. Using the induction hypothesis and saturation up to redundancy, we may infer that $C' \vee A$, and hence $C$, are true in $I^C$. □

A similar lemma can be established for admissible orderings in general, but only for a variant of the ordered chaining calculus with a less restrictive version of composition resolution. As an immediate corollary of the above lemma we obtain the following refutational completeness theorem.

THEOREM 2. *Let $\succ$ be a left-to-right admissible ordering. If a set of clauses $N$*

*is saturated up to redundancy with respect to the chaining calculus $\mathcal{C}^{\succ}_{\mathsf{P}}$, then the set $N \cup \mathsf{CL}_{\mathsf{P}}$ is unsatisfiable if and only if $N$ contains the empty clause.*

## 5. CONGRUENCE RELATIONS

### 5.1 Replacement axioms

Efficient techniques for equational reasoning are indispensable in most theorem proving applications. The chaining techniques proposed in this paper derive from methods that were originally developed for handling equality. We next formulate a "subterm chaining" calculus that combines equational reasoning with chaining. The symbol $\approx$ is used to denote equality.

Equality is a reflexive relation that satisfies all *replacement laws* of the form

$$x \approx y\,,\, A[x] \rightarrow A[y],$$

where the atomic formula $A[y]$ is obtained from $A[x]$ by replacing an indicated occurrence of $x$ by $y$. These axioms express the fact that a formula remains true if a subterm is replaced by an equal subterm. They also imply that the equality relation is transitive and symmetric. In the presence of reflexivity, the replacement laws are also equivalent to the *monotonicity laws*

$$x \approx y\,,\, P(\ldots x \ldots) \;\rightarrow\; P(\ldots y \ldots)$$
$$x \approx y \;\rightarrow\; f(\ldots x \ldots) \approx f(\ldots y \ldots)$$

for all predicate symbols $P$ and function symbols $f$. Thus, if the signature is finite we have a finite representation of the replacement laws (though for our exposition of chaining the finiteness of the representation of the replacement axioms is not essential.) Our intention is to build the monotonicity laws into the chaining calculus via "subterm chaining."

We use the letters $p$ and $q$ to denote *(subterm) positions* in expressions. (Positions may be represented using Dewey decimal notation, for instance.) By $E|_p$ we denote the subterm of an expression $E$ at position $p$, and by $E[u]_p$ the result of replacing in $E$ the subterm at position $p$ by $u$.

To simplify the technical presentation we restrict ourselves from now on to languages with special predicates only.[13] Let $\mathsf{P}$ again be a partially ordered set of predicate symbols, with $\approx$ as the (unique) minimum element. By $\mathsf{EC}_{\mathsf{P}}$ we denote the set of composition laws $\mathsf{CL}_{\mathsf{P}}$, plus the reflexivity and monotonicity laws, for all function and predicate symbols in the language. Symmetry will also be built directly into the notation, in that we use $s \approx t$ to denote both $s \approx t$ and $t \approx s$.[14]

The reflexive-monotonic closure $I^*$ of a Herbrand interpretation $I$ is defined as the set $I$ plus all atoms $u \approx u$ and $u[s]_p \approx u[t]_p$, where $u$ is a ground term, $p$ a position in $u$, and $s \approx t$ an atom in $I$. By the *equational closure* of $I$ (with respect to $\mathsf{EC}_{\mathsf{P}}$) we mean the composition closure of $I^*$ (with respect to $\mathsf{CL}_{\mathsf{P}}$).

[13] Other predicate symbols can be encoded via equality, by representing atoms $P(t_1, \ldots, t_n)$ as equations $P(t_1, \ldots, t_n) \equiv \top$ (using a many-sorted framework).

[14] If $L$ is a literal $u \approx v$, or the negation thereof, we take $a_L$ to be 0, if $u$ and $v$ are different, and 1, if they are the same. This conforms to the definition of $a_L$ in that $L$ does denote a literal with the maximal subterm as the second argument (even though it also denotes a literal with the maximal term as the first argument).

A chain is called an *equational proof* in $I$ if it is a proof in $I^*$. It can easily be shown that a chain

$$(u_0, S_1, u_1), (u_1, S_2, u_2), \ldots, (u_{n-1}, S_n, u_n)$$

is an equational proof in $I$ if, for all $i$ with $1 \leq i \leq n$, either (i) $u_{i-1}\, S_i\, u_i$ is true in $I$, or (ii) $S_i$ is the equality predicate and $u_{i-1} = u_i$, or (iii) there exists a position $p_i$, such that $u_{i-1}|_{p_i} \approx u_i|_{p_i}$ is true in $I$ and $u_{i-1} = u_i[u_{i-1}|_{p_i}]_{p_i}$.

PROPOSITION 3. *A Herbrand interpretation $I$ is a model of* $\mathsf{EC_P}$ *if and only if it is identical to its equational closure.*

PROOF. Use induction on the length of proofs to show that whenever $I$ is a model of $\mathsf{EC_P}$ and $u\, S\, v$ is provable in $I$, then $u\, S\, v$ is true in $I$. For the other direction use the fact that whenever there is a proof of $u \approx v$ in $I$, then there is also a proof of $t[u]_p \approx t[v]_p$ in $I$, for any ground term $t$ and position $p$ in $t$. A proof of $t[u]_p \approx t[v]_p$ is obtained from a proof of $u \approx v$ by simply replacing each term $u_i$ in the chain by the term $t[u_i]_p$. Note that in a proof of type $u \approx v$ only the equality predicate may occur (because it is the minimal predicate) and that all atoms $u \approx u$ are provable. □

Since reflexivity is built into the notion of equational proof, we have the following result about redundancy.

LEMMA 10. *Any clause $C \vee t \approx t$ is* $\mathsf{EC_P}$*-redundant.*

We need to place additional restrictions on the ordering $\succ$, so as to obtain a useful commutation lemma. A *rewrite relation* is a binary relation $\succ$ such that $u \succ v$ implies $E[u]_p \succ E[t]_p$, for all ground expressions $E$, ground terms $u$ and $v$, and positions $p$ in $E$. A *reduction ordering* is a well-founded rewrite relation.

A peak $u \Leftarrow_S t \Rightarrow_T v$ is said to be *critical* if $u\, S\, t$ or $t\, T\, v$ is true in $I$. A plateau $u \Leftrightarrow_S u \Rightarrow_T v$ or $u \Leftarrow_T v \Leftrightarrow_S v$ is called *critical* if $T <_{\mathsf{P}} S$.

LEMMA 11. *Let $\succ$ be a reduction ordering. If all critical peaks and plateaus in $I$ commute, then non-critical peaks and plateaus commute as well.*

PROOF. Suppose all critical peaks and plateaus commmute. We have to show that non-critical peaks and plateaus also commute. A plateau $u \Leftrightarrow_S u \Rightarrow_T v$ or $u \Leftarrow_T v \Leftrightarrow_S v$ is non-critical if $S \leq_{\mathsf{P}} T$, in which case $u \Rightarrow_T v$ and $u \Leftarrow_T v$, respectively, are rewrite proofs of the same type.

A peak $u \Leftarrow_T t \Rightarrow_S v$ is non-critical if both $u\, T\, t$ and $t\, S\, v$ are false in $I$. This implies that both $S$ and $T$ are the equality predicate and that there are (non-root) positions $p$ and $q$, such that (i) $u|_p \approx t|_p$ is true in $I$ and $u = t[u|_p]_p$, and (ii) $t|_q \approx v|_q$ is true in $I$ and $t = v[t|_q]_q$. If $p$ and $q$ denote positions of disjoint subterms in $t$, then there is a rewrite proof $u \Rightarrow_T t' \Leftarrow_S v$, where $t' = (t[u|_p]_p)[v|_q]_q$. Suppose the two subterm positions are not disjoint, say $q = p.q'$. Then there is a critical peak $u|_p \Leftarrow_T t|_p \Rightarrow_S v|_p$ that commutes by assumption. Since $\succ$ is a reduction ordering and $u = t[u|_p]_p = v[u|_p]_p$, the existence of a rewrite proof of $u|_p \approx v|_p$ implies the existence of a rewrite proof of $u \approx v$. That is, the original peak $u \Leftarrow_T t \Rightarrow_S v$ commutes. □

As a corollary we obtain:

LEMMA 12. *The rewrite closure of $I$ with respect to* $\mathsf{EC_P}$ *is a model of* $\mathsf{EC_P}$ *if and only if all critical peaks and plateaus in $I$ commute.*

## 5.2 Subterm chaining

The subterm chaining calculus $\mathcal{SC}^{\succ}_{\mathsf{P}}$, or simply $\mathcal{SC}^{\succ}$, is an extension of the ordered chaining calculus $\mathcal{C}^{\succ}_{\mathsf{P}}$ in which instances of (ordered and negative) chaining with an equality predicate in a side or main literal are generalized to subterm chaining and instances of composition resolution with an equality predicate in the main literal are replaced by equality factoring.

*Ordered Subterm Chaining.*

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

where (i) $\sigma$ is the most general unifier of $s$ and $s'$; (ii) $s\sigma \approx t\sigma$ is strictly maximal with respect to $C\sigma$ and $t\sigma \not\succeq s\sigma$; (iii) $u\sigma\, S\, v\sigma$ is strictly maximal with respect to $D\sigma$ and $v\sigma \not\succ u\sigma$ and, if $S$ is the equality predicate, $v\sigma \neq u\sigma$; (iv) $(s\sigma \approx t\sigma) \not\succeq (u\sigma\, S\, v\sigma)$; and (v) $s'$ is not a variable.

*Negative Subterm Chaining.*

$$\frac{C\,,\, s \approx t \quad D\,,\, \neg(u[s']_p\, S\, v)}{C\sigma\,,\, D\sigma\,,\, \neg(u[t]_p\sigma\; S\, v\sigma)}$$

where (i) $\sigma$ is the most general unifier of $s$ and $s'$; (ii) $s\sigma \approx t\sigma$ is strictly maximal with respect to $C\sigma$ and $t\sigma \not\succeq s\sigma$; (iii) $\neg(u\sigma\, S\, v\sigma)$ is maximal with respect to $D\sigma$ and $v\sigma \not\succeq u\sigma$; and (iv) $s'$ is not a variable.

and

$$\frac{C\,,\, s \approx t \quad D\,,\, \neg(u\, S\, v[s']_p)}{C\sigma\,,\, D\sigma\,,\, \neg(u\sigma\, S\, v[t]_p\sigma)}$$

where (i) $S$ is not the equality predicate[15] and $\sigma$ is the most general unifier of $s$ and $s'$; (ii) $s\sigma \approx t\sigma$ is strictly maximal with respect to $C\sigma$ and $t\sigma \not\succeq s\sigma$; (iii) $\neg(u\sigma\, S\, v\sigma)$ is maximal with respect to $D\sigma$ and $u\sigma \not\succ v\sigma$; and (iv) $s'$ is not a variable.

*Reflexivity Resolution.*

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

where $\sigma$ is the most general unifier of $u$ and $v$, and $u\sigma \not\approx v\sigma$ is maximal with respect to $C\sigma$.

*Equality Factoring.*

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

[15] Since the equality predicate is symmetric, the first (more restrictive) variant of negative chaining is sufficient.

where (i) $\sigma$ is the most general unifier of $u$ and $u'$; (ii) $u'\sigma \approx v'\sigma$ is strictly maximal with respect to $C\sigma \vee u\sigma \approx v\sigma$; and (iii) $v\sigma \not\succeq u\sigma$, $v'\sigma \not\succeq u'\sigma$, and $v\sigma \not\succeq v'\sigma$.

Equality factoring corresponds to composition resolution of a certain clause with itself. The following lemma can be proved in a similar way as Lemma 8. (Note that if $\succ$ is an admissible reduction ordering, then $u[t] \succ t$, for all ground terms $u$ and proper subterms $t$ of $u$.)

LEMMA 13. *Ground inference rules of the subterm chaining calculus are decreasing as follows:*

*(1) A ground inference by negative subterm chaining, reflexivity resolution, or equality factoring is decreasing if the reduction ordering $\succ$ is admissible.*

*(2) A ground ordered subterm chaining inference is decreasing if $\succ$ is left-to-right admissible.*

### 5.3 Lifting

A key issue with subterm chaining is whether chaining inferences at, or even below, variable positions are required. Such inferences would not be feasable in practice, as non-ground clauses by definition contain variables as subterms. Fortunately, in the case of symmetric relations, such as equality, chaining at or below variables is not needed:

LEMMA 14. *Let*

$$\frac{C\,,\, s \approx t \quad D\,,\, L[s]_p}{C\,,\, D\,,\, L[t]_p}$$

*be a ground inference by ordered or negative subterm chaining, where the second premise is a ground instance $D'\sigma \vee L'\sigma$ of some clause in $N$ and $p = q.q'$, with $q$ being a position in $L'$ such that $L'|_q$ is a variable. Then the inference is $\mathsf{EC_P}$-redundant in $N$.*

PROOF. We have $s \succ t$ and $s \approx t$ is the side literal of the inference. Suppose $q$ is a position in $L'$ such that $p = q.q'$ and $L'|_q = x$, for some variable $x$. Furthermore, suppose $x\sigma = u$ and $u|_{q'} = s$. Let $\tau$ be the same substitution as $\sigma$ except that $x\tau = u[t]_{q'}$. Then $D'\tau \vee L'\tau$ is a ground instance of $N$ that is smaller than $D'\sigma \vee L'\sigma$, and the latter clause is true in every $\mathsf{EC_P}$-model of $D'\tau \vee L'\tau$ and $s \approx t$. Therefore the above inference is $\mathsf{EC_P}$-redundant in $N$. $\square$

A consequence of this lemma is that any non-redundant inference from ground instances of $N$ can be obtained as ground instance of an inference from premises in $N$.

PROPOSITION 4. *Let $N$ be a set of clauses that is saturated up to redundancy with respect to $\mathcal{SC}$. Then the set of ground instances of $N$ is also saturated up to redundancy with respect to $\mathcal{SC}$.*

### 5.4 Refutational completeness

We have to slightly modify our model construction for equational proofs, which require the use of reflexive-monotonic closures. 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 $I_C$ the rewrite closure of

$R^*_C$. Furthermore, if $C$ is a clause $C' \vee A$, where $A$ is a positive literal with $A \succ C'$, and $C$ is false in $I_C$ then $E_C = \{A\}$. In that case, we also say that $C$ is *productive* (and *produces* $A$). In all other cases, $E_C = \emptyset$. Finally, let $R$ be the set $\bigcup_C E_C$ and $I$ the rewrite closure of $R^*$. The set $R^C$ is defined as before and by $I^C$ we denote the rewrite closure of $(R_C \cup E_C)^*$.

Lemmas 3, 4, and 5 carry over to this modified model construction. We have the following main lemma:

LEMMA 15. *Let $\succ$ be a left-to-right admissible reduction ordering and $N$ be a set of clauses that is saturated up to $\mathsf{EC_P}$-redundancy with respect to $\mathcal{SC}^{\succ}_{\mathsf{P}}$ and does not contain the empty clause. In addition, 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 not $\mathsf{EC_P}$-redundant in $N$.*

*(2) If $C = C' \vee A$ produces $A$ and $D = D' \vee u \approx v$ produces $u \approx v$, where $C \succ D$ and $u \succ v$, then $u$ is not a subterm of $A$.*

*(3) If $D = D' \vee A$ produces $A$ and $C \succeq D$, then $D'$ is false in $I^C$.*

*(4) The interpretations $I_C$ and $I_C$ are models of $\mathsf{EC_P}$.*

*(5) $C$ is true in $I^C$.*

PROOF. The proof is by induction on $\succ$. By Proposition 4 we may assume that the set of ground instances of $N$ is saturated up to redundancy. Let $C$ be a ground clause in $N$, such that assertions (1)–(5) are satisfied for all smaller clauses in $N$.

(1) The proof of this part is virtually the same as for Lemma 9.

(2) Let $C$ and $D$ be clauses as indicated and suppose $u$ is a subterm of $A$. Then the clause $C'' = C' \vee D' \vee A[v]$ can be obtained from $C$ and $D$ by ordered subterm chaining. The side literal $L$ of the inference is $u \approx v$. By part (1) both $C$ and $D$ are non-redundant. Since the set of all ground instances of $N$ is saturated up to redundancy, there exist ground instances $C_1, \ldots, C_n$ of clauses in $N$, all of which are smaller than $C$, such that $C''$ is true in every $\mathsf{EC_P}$-model of $\{L, C_1, \ldots, C_n\}$. We may use Lemma 4 and parts (4) and (5) of the induction hypothesis to infer that $I_C$ is one such model. Thus the clause $C''$ is true in $I_C$. The subclause $C'$ is false in $I_C$ and, by part (3) of the induction hypothesis, $D'$ is also false in $I_C$. Therefore, $A[v]$ must be true in $I_C$. But if there is a rewrite proof of $A[v]$ in $R_C$, then there is also a rewrite proof of $A = A[u]$, because $u \succ v$ and $u \approx v$ is an element of $R_C$. This contradicts the fact that $C$ is productive.

(3) This part can be proved basically in the same way as in the proof of Lemma 9. The main difference is that we have to show that inferences by composition resolution,

$$\frac{D'', u \approx v, u \approx w \quad C', u \approx u'}{C', v \not\approx u', u \approx v,}$$

in which the main literal is an equality, are not needed. We know that such an inference is only needed when both premises are productive clauses and there is a rewrite proof of $v \approx u'$ in $R_C$, where $C$ denotes the main premise. Suppose there is such an inference. Since the maximal term in both premises is the same (namely $u$), we may use part 2 to infer that the two premises are identical, i.e., $w = u'$ and hence there is a rewrite proof of $v \approx w$ in $R_C$. Now consider the equality factoring inference

$$\frac{D'', u \approx v, u \approx w}{D'', v \not\approx w, u \approx v}$$

with premise $C$. We may use parts (4) and (5) of the induction hypothesis and the fact that the set of all ground instances of $N$ is saturated up to redundancy, to infer that the conclusion of this inference is true in $I_C$. Since $C$ is productive, the subclause $D''$, $u \approx v$ must be false in $I_C$. But then $v \not\approx w$ has to be true in $I_C$, which contradicts the fact that there is a rewrite proof of $v \approx w$ in $R_C$.

Parts (4) and (5) are proved in a similar way as the corresponding statements in Lemma 9. □

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

THEOREM 3. *If $\succ$ is a left-to-right admissible reduction ordering and a set of clauses $N$ is saturated up to redundancy with respect to $\mathcal{SC}_{\mathsf{P}}^{\succ}$, then the set $N \cup \mathsf{EC}_{\mathsf{P}}$ is unsatisfiable if and only if it contains the empty clause.*

### 5.5 Partial congruences

A *partial equivalence* $\sim$ is a symmetric and transitive (but not necessarily reflexive) binary relation. It is called a *partial congruence* if it satisfies the replacement axioms

$$x \sim y,\ A[x] \ \rightarrow\ A[y]$$

for all atoms $A$ and all occurrences of $x$ in $A$. A partial congruence need not satisfy any of the monotonicity laws, but monotonicity (together with symmetry and transitivity) implies the replacement axioms.

Partial congruences are useful for axiomatizing partial functions. They occur in Scott's logic of partial equality [Scott 1979] and also in type theory where $t \approx_T t$ in a type $T$ can be proven only if $t$ is provably a member of $T$.

For example, the natural numbers with a partial predecessor function can be specified as follows:

$$\begin{aligned}
nat(x) &\leftrightarrow x \sim x \\
&\rightarrow nat(0) \\
nat(x) &\leftrightarrow nat(s(x)) \\
0 \sim s(x) &\rightarrow \\
nat(x) &\rightarrow p(s(x)) \sim x \\
nat(p(x)) &\rightarrow s(p(x)) \sim x
\end{aligned}$$

The undefinedness of $p(0)$ is represented by the fact that $\neg nat(p(0))$ can be derived as a theorem.

An example of a partial congruence for which the replacement axioms can be represented finitely (for finite signatures) is equality for strict partial functions:

$$\begin{aligned}
x \sim y &\rightarrow y \sim x \\
x \sim y, y \sim z &\rightarrow x \sim z
\end{aligned}$$

$$f(x_1, \ldots, x_n) \sim f(x_1, \ldots, x_n) \;\rightarrow\; x_i \sim x_i, \text{ for any } f \text{ and position } i$$
$$x_i \sim y_i, f(\ldots x_i \ldots) \sim z \;\rightarrow\; f(\ldots y_i \ldots) \sim z, \text{ for any } f \text{ and } i$$

It is easy to see that these axioms in fact imply the replacement axioms. The third axiom (scheme) specifies strictness of all functions.

Symmetry axioms, $x\,S\,y \rightarrow y\,S\,x$, for special predicates can be easily built into the chaining mechanism by extending the notation in the same way as for equality. That is, an expression $s\,S\,t$ denotes both $s\,S\,t$ and $t\,S\,s$, whenever $S$ denotes a symmetric relation. Note that if a predicate $S$ is symmetric and $T <_{\mathsf{P}} S$, then $T$ should also be symmetric, so that the notion of proof we have used is adequate. (If all the predicates occurring in a proof of $u\,S\,v$ are symmetric, then one can obtain a proof of $v\,S\,u$ by simply reverting all proof steps in the original proof.)

The composition laws and symmetry also imply "partial reflexivity" properties. In particular, we obtain implications

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

whenever $\sim$ is symmetric. Call a term *defined* (with respect to $\sim$) if $t \sim t$ can be derived; and the set of all defined ground terms, the *domain* of the partial equivalence. The above implications allow one to infer "definedness" of terms. They can be built into the inference mechanism by corresponding inference rules:

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

where $u\sigma\,S\,v\sigma$ is maximal with respect to $C\sigma$ and $v\sigma \not\succeq u\sigma$. For further details on subterm chaining for partial equivalence and congruence relation see Bachmair and Ganzinger [1995].

### 5.6 Monotonicity and anti-monotonicity

The subterm chaining calculus employs standard rewriting techniques for treating replacement axioms. In practice, one often encounters theories with non-symmetric relations that satisfy monotonicity properties. For example, to define ordered ring structures one needs axioms such as

$$x < y \;\rightarrow\; x + z < y + z$$
$$x < y \;\rightarrow\; -y < -x$$

that express monotonicity and anti-monotonicity, respectively, of certain functions with respect to a (non-symmetric) partial order. 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 completeness requires chaining into variables of the functional-reflexive axioms (cf. the "variable instance pairs" in Bachmair, Dershowitz, and Hsiang [1986]), 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 have an alternative rewrite proof

$$a < b < b * b < a * b$$

of the same type, so that the chaining inference would be unnecessary. This example, which illustrates the problems caused by variable overlaps, depends on the syntactic ordering $\succ$ and on the non-linearity of the variable $x$.

In the case of non-unit clauses, a proper handling of negative literals poses further difficulties. Take the two clauses

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

plus the monotonicity axiom $x < y \rightarrow f(x) < f(y)$. This set of clauses is unsatisfiable: from the first clause and the monotonicity axiom we get $f(f(a)) < f(f(b))$, which contradicts the second clause. However, we cannot get a contradiction from the two clauses by chaining even at variable subterms, regardless of how the ordering is defined. The difficulty is that the unifier that is obtained via the chaining inference will not yield the right instantiation for the variable $x$. A contradiction can be obtained if we combine chaining into a variable with a subsequent unification of the two terms in the resulting negative literal. Thus, we get $\neg(f(x) < f(f(a)))$ as an intermediate result 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*, as we get the same effect by applying the "context" $f(f(.))$ to both terms in $a < b$, and then resolving with $\neg(f(x) < f(f(y))$. 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 monotonicity 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. But variable chaining provides no clue about the context in this case.

No satisfactory general solutions to this problem have been proposed yet. Monotonicity axioms cannot be fully captured by subterm chaining without chaining below variables. Monotonicity axioms may be dealt with by standard ordered chaining, but perhaps with a more refined analysis of redundancy for especially prolific inferences. For instance, ordered chaining inferences

$$\frac{x < y \rightarrow f(x) < f(y) \quad y < z \rightarrow f(y) < f(z)}{x < y\,,\, y < z \rightarrow f(x) < f(z)}$$

where $<$ is transitive, are redundant in any set $N$ that contains the clause $x < y \rightarrow f(x) < f(y)$. The ordering restrictions on the inference imply that $y \succ x$ and $y \succ z$. To show redundancy we have to find a proof of the conclusion that uses only the transitivity law and instances of clauses in $N$ that are smaller than the maximal premise $y < z \rightarrow f(y) < f(z)$. Clearly, $x < y \wedge y < z$ implies $x < z$ by transitivity and any instance of $x < z \rightarrow f(x) < f(z)$ is smaller than the corresponding instance of $y < z \rightarrow f(y) < f(z)$ whenever $y \succ x$.[16]

[16] We assume that $\succ$ is a reduction ordering. This is of advantage, even if one does not employ

It seems possible to develop special versions of subterm chaining for particular algebraic structures such as ordered groups and fields or lattices, as the richer algebraic structure often simplifies the problem considerably (though the technical treatment may get much more involved). Typically, one has to use extended rewriting modulo associativity and commutativity and integrate specialized simplification techniques afforded by the given algebraic structure. Examples of this approach are the specialized versions of the associative-commutative superposition calculus for abelian groups [Waldmann 1998] and for commutative rings without orderings [Bachmair et al. 1995].

## 6. TOTAL ORDERINGS

We next present a refinement of the subterm chaining calculus for total orderings. Let $\succ$ be an admissible ordering and $\mathsf{P}$ be a set of special predicates, containing two distinguished symbols $\approx$ and $<$ meant to denote the equality relation and a total ordering, respectively.[17] We assume that $<_{\mathsf{P}}$ is an admissible ordering on $\mathsf{P}$ in which $\approx$ is the (unique) minimum and the symbol $<$ is comparable to no symbol other than equality. We call an atom $s < t$ an *inequality* and write $s \not< t$ to denote the negative inequality $\neg(s < t)$. The properties of a total ordering are specified by an irreflexivity axiom,

$$x \not< x,$$

and a totality axiom,

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

These axioms will be considered along with the composition laws $\mathsf{EC_P}$ (which include the transitivity law for $<$).

We first examine the various inferences in which irreflexivity and totality axioms can participate and outline how to organize the saturation process so as to minimize the number of inferences with these axioms. In particular, we show that composition resolution inferences with inequalities can be drastically simplified. Finally, we will again address the problem of variable chaining and present simplification techniques whereby one may avoid most, sometimes all, variable chaining inferences.

### 6.1 Ordered chaining for total orderings

The irreflexivity axiom may occur as the second premise in ordered resolution and negative chaining inferences. (Since the axiom contains only variable subterms, it cannot be used as the main premise in negative *subterm* chaining inferences.) We first reformulate ordered resolution with the irreflexivity axiom as a (single-premise) inference rule.

*Irreflexivity Resolution.*

$$\frac{C, u < v}{C\sigma}$$

---

subterm chaining, because it allows for stronger redundancy arguments.

[17] The ordering $<$ (on the object level) should not be confused with the term ordering $\succ$ (on the meta-level) that is used to restrict the search space for theorems about $<$.

where (i) $\sigma$ is the most general unifier of $u$ and $v$; and (ii) $u\sigma < v\sigma$ is strictly maximal with respect to $C\sigma$.

In the presence of irreflexivity resolution negative chaining inferences with the irreflexivity axiom turn out to be redundant:

PROPOSITION 5. *Let $\succ$ be an admissible ordering and $N$ be a set of clauses containing the irreflexivity axiom, such that all inferences by irreflexivity resolution from $N$ are $\mathsf{EC_P}$-redundant in $N$. Then any inference by negative chaining with $x \not< x$ is $\mathsf{EC_P}$-redundant in $N$.*

PROOF. Consider a ground negative chaining inference

$$\frac{C\,,\, u < v \quad u \not< u}{C\,,\, \neg(v\, S\, u)}$$

where $u \succ v$ and $(u < v) \succ C$ (and $S$ is either $\approx$ or $<$). The side literal of this inference is $u < v$. We prove that the conclusion $C \vee \neg(v\, S\, u)$ is true in every $\mathsf{EC}$-model of $u < v$ and $v \not< v$. Let $I$ be such a model. Suppose $C$ is false and $v\, S\, u$ true in $I$. We may use the composition laws and the fact that $u < v$ is true in $I$, to infer that $v < v$ is also true in $I$. This contradicts the assumption that $I$ is a model of $v \not< v$. We conclude that $C \vee \neg(v\, S\, u)$ is true in $I$.

Similar arguments can be applied to the second kind of negative chaining inferences. □

Let us next look at possible inferences with the totality axiom. Again we first formulate a corresponding one-premise resolution rule.

*Totality Resolution.*

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

where $u \neq v$.

This inference rule covers ordered resolution with the totality axiom, but also "non-ordered" (and hence redundant) resolution inferences in which the resolved literals are not maximal in their respective premises. We prefer the "partially redundant" inference rule because it has the interesting property that the conclusion renders the premise redundant.

PROPOSITION 6. *Let $\succ$ be an admissible ordering and $N$ be a set of clauses containing the irreflexivity axiom. A clause $C \vee u \not< v$ is either an instance of the irreflexivity axiom or else is $\mathsf{EC_P}$-redundant in $N \cup \{\, C \vee v < u \vee v \approx u \,\}$.*

PROOF. It is sufficient to prove the assertion for all *ground* clauses $C \vee u \not< v$. If $u = v$, then the clause is subsumed by an instance of the irreflexivity axiom. Thus, if it is not an instance of the axiom, then it is redundant.

Suppose $u$ and $v$ are distinct terms. We consider the case $u \succ v$ (the case $v \succ u$ is proved in a similar way). First note that $C \vee v < u \vee v \approx u$ is a smaller clause than $C \vee u \not< v$ in any admissible ordering. Let $I$ be an $\mathsf{EC_P}$-model of $C \vee v < u \vee v \approx u$ and $v \not< v$. If $C$ is false in $I$, then either $v < u$ or $v \approx u$ must be true in $I$. If in addition $u < v$ were true in $I$, then we could use composition to infer that $v < v$ is also true in $I$, which would contradict that $I$ is a model of $v \not< v$. We conclude that

$u \not< v$, and hence $C \vee u \not< v$, is true in $I$, and have thus proved that $C \vee u \not< v$ is $\mathsf{EC_P}$-redundant as claimed. □

Inference rules in which the conclusion renders one of the premises redundant are called *simplification rules*. They allow one to *replace* the redundant premise by the conclusion. Thus totality resolution allows us to eliminate virtually all clauses with negative inequalities. The only clauses with a negative inequality that can not be eliminated by totality resolution are subsumed by the irreflexivity axiom. If a set contains no clauses with negative inequalities (other than the irreflexivity law), then obviously no negative (subterm) chaining inferences with inequalities are needed.

Let us next take a look at ordered (subterm) chaining inferences with the totality axiom. Again, observe that there are no subterm chaining inferences with this axiom, as it contains only variable subterms. Now take a ground instance $u < v \vee v < u \vee u \approx v$. If $u$ and $v$ are identical, then the clause is redundant by Lemma 10. If the two terms are distinct, and the given ordering $\succ$ is left-to-right admissible, then the maximal literal of the clause is the one with the maximal term as the first argument, which means that such an instance of the axiom can only occur as the second premise of an ordered (subterm) chaining inference. Such inferences are redundant:

PROPOSITION 7. *If $\succ$ is a left-to-right admissible ordering, then any ground ordered chaining inference*

$$\frac{C\,,\, u < v \qquad v < w\,,\, w < v\,,\, v \approx w}{C\,,\, u < w\,,\, w < v\,,\, v \approx w}$$

*is $\mathsf{EC_P}$-redundant in any set $N$ that contains the totality axiom.*

PROOF. A chaining inference as specified is ordered if $v \succ u$, $v \succ w$ and $(u < v) \succ C$. The side literal of the inference is $u < v$. Let $C'$ denote the conclusion of the inference. It suffices to show that $C'$ is true in every $\mathsf{EC_P}$-model of $u < v$ and $u < w \vee w < u \vee u \approx w$. Let $I$ be such a model. If $u < w$ is true in $I$, then obviously $C'$ is also true in $I$. On the other hand, if $w < u \vee w \approx u$ is true in $I$, then using composition and the fact that $u < v$ is true in $I$, we may infer that $w < v$ is true in $I$, which again implies that $C'$ is true in $I$. □

The following proposition suggests possible optimizations of composition resolution for inequalities.

PROPOSITION 8. *Let $\succ$ be an admissible ordering and $N$ be a set of clauses containing the irreflexivity axiom and $C$ be a ground clause $C' \vee s < t \vee s < u$ with maximal term $s$ and $s \succ t$ and $s \succ u$. Then $C$ is $\mathsf{EC_P}$-redundant in*

$$N \cup \{\, C' \vee s < u \vee u < t \vee u \approx t,\ C' \vee s < t \vee t < u \,\}.$$

*Similarly, if $C$ is of the form $C' \vee t < s \vee u < s$ with maximal term $s$ and $s \succ t$ and $s \succ u$, then $C$ is $\mathsf{EC_P}$-redundant in*

$$N \cup \{\, C' \vee u < s \vee t < u \vee t \approx u,\ C' \vee t < s \vee u < t \,\}.$$

PROOF. For the first case observe that the clauses $C' \vee s < u \vee u < t \vee u \approx t$ and $C' \vee s < t \vee t < u$ and the unit clause $t \not< t$ are all smaller than $C$. We claim that $C$ is true in any $\mathsf{EC_P}$-model $I$ of these three clauses. For if $C'$ and $s<t$ and $s<u$ are all false in $I$, then $u<t \vee u \approx t$ and $t<u$ must be true. We may the use composition to infer that $t<t$ is true in $I$, which is a contradiction. The second case is symmetric to this case. □

Note that by composition resolution of $C \vee s < t \vee s < u$ with itself we can obtain both $C \vee s < u \vee t \not< u$ and $C \vee s < u \vee t \not\approx u$. From the first of these two clauses we get $C \vee s < u \vee u < t \vee u \approx t$ by totality resolution; from the second, $C \vee s < t \vee t < u$ by (non-ordered) resolution with totality axiom, followed by chaining. These observations motivate the following inference rules, which will replace those instances of composition resolution in which the main literal is an inequality.

*Inequality Factoring.*

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

and

$$\frac{C\,,\, s<t\,,\, s'<u}{C\sigma\,,\, s\sigma<t\sigma\,,\, t\sigma<u\sigma}$$

where (i) $\sigma$ is the most general unfier of $s$ and $s'$, (ii) $s\sigma<t\sigma$ is strictly maximal in $C\sigma \vee s\sigma<u\sigma$, and (iii) $t\sigma \not\succeq s\sigma$ and $u\sigma \not\succeq s'\sigma$; and and

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

and

$$\frac{C\,,\, t<s\,,\, u<s'}{C\sigma\,,\, t\sigma<s\sigma\,,\, u\sigma<t\sigma}$$

where (i) $\sigma$ is the most general unfier of $s$ and $s'$, (ii) $t\sigma<s\sigma$ is strictly maximal in $C\sigma \vee u\sigma<s\sigma$. and (iii) $t\sigma \not\succeq s\sigma$ and $u\sigma \not\succeq s'\sigma$.

LEMMA 16. *If $\succ$ is an admissible ordering, then any ground inference by irreflexivity or totality resolution or by inequality factoring is decreasing with respect to $\succ$.*

Let now $\mathcal{O}^{\succ}_{\mathsf{P}}$ be the inference system $\mathcal{SC}^{\succ}_{\mathsf{P}}$ plus irreflexivity resolution, totality resolution and inequality factoring, but without inference rules in which the irreflexivity or totality axiom is a premise and without any instances of composition resolution in which the main literal is an inequality.

We obtain the following corollary to Theorem 3.

THEOREM 4. *Let $\succ$ be a left-to-right admissible ordering and $N$ be a set of clauses containing the irreflexivity and totality axioms, but no clauses (other than the irreflexivity axiom) with negative inequalities. If $N$ is saturated up to $\mathsf{EC_P}$-redundancy with respect to $\mathcal{O}^{\succ}_{\mathsf{P}}$, then $N \cup \mathsf{EC_P}$ is unsatisfiable if and only if it contains the empty clause.*

PROOF. Suppose $N$ is saturated up to $\mathsf{EC_P}$-redundancy with respect to $\mathcal{O}^{\succ}_{\mathsf{P}}$. By Theorem 3 it suffices to show that $N$ is saturated up to $\mathsf{EC_P}$-redundancy with respect to $\mathcal{SC}^{\succ}_{\mathsf{P}}$.

One difference between the two calculi, $\mathcal{O}_{\mathsf{P}}^{\succ}$ and $\mathcal{SC}_{\mathsf{P}}^{\succ}$, is that in the former there are no inferences with the irreflexivity or totality axiom as a premise. Irreflexivity and totality resolution cover ordered resolution inferences with these two axioms. By Proposition 5 any negative chaining inference with the irreflexivity axiom is redundant. Similarly, by Proposition 7 any ordered chaining inference with the totality axiom (that is possible with a left-to-right admissible ordering) is redundant. There are no subterm chaining inferences with the two axioms, as they contain only variables subterms. In short, all inferences in the calculus $\mathcal{SC}_{\mathsf{P}}^{\succ}$ that involve the the irreflexivity or totality axiom are redundant.

The other difference between the two calculi is that inequality factoring replaces composition resolution for inequalities. The first premise of any ground composition resolution inference for inequalities is of the form $C \vee s < t \vee s < u$. If $s = t$ or $s = u$, then the clause is redundant, as the set $N$ is saturated up to redundancy with respect to irreflexivity resolution. If $s \succ t$ and $s \succ u$, we may use Proposition 8 and the fact that $N$ is saturated up to redundancy with respect to inequality factoring to infer that the clause is also redundant. This implies that such inferences by composition resolution are redundant.

In sum, all inferences in $\mathcal{SC}_{\mathsf{P}}^{\succ}$ from clauses in $N$ are redundant and therefore the set $N$ is indeed saturated up to redundancy with respect to $\mathcal{SC}_{\mathsf{P}}^{\succ}$. □

## 6.2 Non-strict orderings

When we use totality resolution to convert negative inequalities $s \not< t$ to disjunctions $t < s \vee t \approx s$, the resulting duplication of terms may double the size of a clause. Chaining inferences may produce even longer clauses with many repeated occurrences of terms. Let us briefly sketch techniques for dealing with this problem. We introduce a meta-level symbol $\leq$ and use $s \leq t$ as an abbreviation for the disjunction $s < t \vee s \approx t$. (In the disjunction the inequality is maximal in any admissible ordering). We may then formulate inference rules directly in terms of *non-strict* inequalities $u \leq v$ and thereby avoid a considerable amount of term duplication.

For instance, the original chaining inference with the expanded versions of the clauses $C \vee s < t$ and $D \vee t \leq u$ yields $C \vee D \vee s < u \vee t \approx u$. Instead we may use a modified inference with conclusion $C \vee D \vee s < u$, which is also sound in the presence of $\mathsf{EC_P}$. Similarly, instead of inferring $C \vee D \vee s < u \vee t \approx u \vee s \approx t$ from $C \vee s \leq t$ and $D \vee t \leq u$, we may deduce $C \vee D \vee s < u \vee t \approx u$. From the two clauses $C \vee s \approx t$ and $D \vee t \leq u$ we get $C \vee D \vee s < u \vee t \approx u$ by ordered chaining, whereas it is also sound to deduce $C \vee D \vee s \leq u$. In interpretations in which (the side literal) $s \approx t$ is true, the two clauses are equivalent but the latter is smaller than the former whenever $t \succ s$. Thus, the old inference is redundant whenever the new inference is redundant.

Inequality factoring can also be changed to avoid term duplication. Examples of alternative (ground) inferences are

$$\frac{C,\, s < t,\, s \leq u}{C,\, t \leq u,\, s < t}$$

and

$$\frac{C,\, s < t,\, s \leq u}{C,\, u < t,\, s \leq u}.$$

### 6.3 Variable elimination

Let us next discuss the issue of variable chaining in the context of total orderings. We assume that clauses may contain both strict and non-strict inequalities, and use the symbol $\lhd$ to denote $<$ or $\leq$, and the symbol $\unlhd$ to denote $<$ or $\leq$ or $\approx$.

Recall that a chaining inference

$$\frac{C\,,\, s \lhd_1 t \quad D\,,\, u \lhd_2 v}{C\sigma\,,\, D\sigma\,,\, s\sigma \lhd t\sigma}$$

where $t\sigma = u\sigma$, is called a *variable chaining* if one of the terms $t$ or $u$ is a variable. We have already pointed out in Section 3.7 that if one chooses a suitable ordering $\succ$ (so that a term is always bigger than all of its proper subterms), then the side conditions of ordered chaining can only be satisfied if each term $t$ or $u$ is either a non-variable or an *unshielded variable*; that is, a variable that does not occur as a nested subterm in its clause. For example, the variable $x$ is unshielded in $a < x \vee x < b$, where $a$ and $b$ are constants; whereas $x$ and $y$ are both shielded in the clause $x \leq y \vee s(y) < s(x)$.

Unshielded variables can often be eliminated from clauses. For instance, for any total ordering, $a < x \vee x < b$ is equivalent to $a < b$; for an ordering without a left endpoint, a disjunction $C \vee \bigvee_i x \approx u_i \vee \bigvee_j t_j < x$, where $x$ does not occur in $C$ or any term $t_j$, is equivalent to $C$; and for a total "discrete" ordering, where $s(x)$ denotes the "successor" of $x$, the disjunction $a \leq x \vee x \leq b$ is equivalent to $a \leq s(b)$.

We will investigate the case of (total) dense orderings without endpoints, the properties of which can be expressed by a set $\mathsf{Do}(d, l, r)$ of four clauses:

$$\begin{gathered} x < d(x,y)\,,\, y \leq x \\ d(x,y) < y\,,\, y \leq x \\ l(x) < x \\ x < r(x) \end{gathered}$$

The following inference rule is sound for total dense orderings without endpoints.

*Variable Elimination.*

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

with $n, m \geq 0$, where $x$ is an unshielded variable not occurring in $C$, $u_i$ and $v_i$, and where $\lhd_{i,j}$ is the minimum of $\lhd_i$ and $\lhd'_j$.[18]

This variable elimination rule has been used by Bledsoe, Kunen, and Shostak [1985] in their chaining calculus. (This calculus does not have an explicit equality predicate, but uses both strict and non-strict inequality predicates at the object level, so that an equality $s \approx t$ can be implicitly represented as a conjunction of two non-strict inequalities $s \leq t$ and $t \leq s$. Refutational completeness in that context requires inferences with congruence axioms, as specified by clauses

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

[18] Here $\leq$ is taken to be smaller than $<$, but bigger then $\approx$.

where $f$ ranges over all function symbols.) Various other variable elimination rules have been proposed in the theorem proving literature; e.g., Hines [1994] describes related elimination rules for a certain fragment of the first-order theory of integers. The above elimination rule is a simplification rule in our sense (the premise is rendered redundant by the conclusion) for suitable orderings.

When reasoning about variable elimination we have to take into account how a ground clause is obtained from a non-ground clause by instantiation. Let $L$ be a ground literal $A$ or $\neg A$ in a ground instance $C\sigma$ of a clause $C$, and let $n_C$ be the number of different variables occuring as arguments of an equality or an 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 $n_L$ to be 0, if $A$ is not an inequality; to be $-1$, if $A$ is an inequality, but $C$ is the totality axiom; and to be $n_C$, otherwise.

We say that an ordering $\succ$ (more precisely, an ordering on occurrences of literals in ground instances of clauses) is *admissible for variable elimination* (in inequalities) if it is left-to-right admissible and if $L \succ L'$ whenever $L$ and $L'$ are *positive* inequalities and the the tuple $(max_{L'}, p_{L'}, s_{L'}, a_{L'}, n_{L'})$ associated with $L'$ is strictly smaller than the tuple $(max_L, p_L, s_L, a_L, n_L)$ associated with $L$. (Tuples are compared lexicographically, using the same orderings as before in the first four components and the usual greater-than relation on the integers in the last component.)

LEMMA 17. *Let $\succ$ be an ordering that is admissible for variable elimination.*

*(i) A clause $C'$ of the form $C \vee \bigvee_i x \trianglelefteq_i u_i$ or $C \vee \bigvee_i u_i \trianglelefteq_i x$, where $x$ is an unshielded variable which does not occur in $C$, is $\mathsf{EC_P}$-redundant with respect to any set of clauses that contains $C$.*

*(ii) Let $C'$ be a clause $C \vee \bigvee_i u_i \lhd_i x \vee \bigvee_j x \lhd_j v_j$, where $x$ does not occur in $C$ or any term $u_i$ or $v_i$, and let $C''$ be $C \vee \bigvee_{i,j} u_i \lhd_{i,j} v_j$, where $\lhd_{i,j}$ is the minimum of $\lhd_i$ and $\lhd'_j$. Then $C'$ is either the totality axiom or else is $\mathsf{EC_P}$-redundant with respect to any set $N$ that contains $C''$ and the totality axiom.*

PROOF. (i) The clause $C'$ is properly subsumed by $C$ and therefore is redundant.

(ii) Let $C'$ and $C''$ be clauses as indicated and suppose that $C'$ is not a totality axiom. Let $D' = D \vee \bigvee_i a_i \lhd_i c \vee \bigvee_j c \lhd_j b_j$ and $D'' = D \vee \bigvee_{i,j} a_i \lhd_{i,j} b_j$ be ground instances of $C'$ and $C''$, respectively. Let $L''$ be a literal $a_i \lhd_{i,j} b_j$ and let $L'$ be the literal $a_i \lhd_i c$, if $a_i \succeq b_j$, and the literal $c \lhd_j b_j$, otherwise. If $c$ is the unique maximal term in $L'$, then evidently $L' \succ L''$. But even if $c$ is not a maximal term in $L'$, we have $L' \succ L''$, because in that case $s_{L''} \leq_{\mathsf{P}} s_{L'}$ and $n_{L'} > n_{L''}$. (The latter relation is true because $C''$ does not contain the unshielded variable $x$, but otherwise contains the same terms as $C'$.) As a consequence, we have $D' \succ D''$.

Let $T_a$ be the set of all ground instances $c < a_i \vee a_i \leq c$ of the totality axiom and $T_b$ be the set of all ground instances $c < b_j \vee b_j \leq c$. The clause $D'$ is true in every $\mathsf{EC_P}$-model of $T_a \cup \{D''\}$ and in every $\mathsf{EC_P}$-model of $T_b \cup \{D''\}$. 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 implies that all clauses in $T_b$ are smaller than $D'$. (Note that $n_L = -1$ for all literals $L$ in $T_b$, which is essential in case $c$ is maximal.) In sum, $D'$ is $\mathsf{EC_P}$-redundant with respect to any set $N$ that contains $C''$ and the totality axiom. □

The lemma indicates that variable elimination is indeed a simplification rule (in the presence of the totality axiom) and that therefore most unshielded variables in (in)equalities can be eliminated. This implies that for sets of clauses in which equality occurs only implicitly, as part of non-strict inequalities $u \leq v$, variable chaining is not needed. Corresponding results have been established for non-ordered chaining calculi, see Richter [1984] and Hines [1992].

For arbitrary equations variable elimination is more complicated. Lemma 17 is applicable, but does not cover all coccurrences of unshielded variables in equalities $x \approx t$ (or negations thereof). We eliminate such unshielded variables by first translating (positive and negative) equalities with $x$ into inequalities, and then applying the 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. We may also assume that $x$ occurs either in some negative equation $x \not\approx u_i$ in $D$ or else in inequalities $x \lhd s$ *and* $t \lhd x$ in $C$, for otherwise Lemma 17 applies. The above formula can be translated into $2^n$ clauses, to each of which the variable elimination rule can be applied. We denote by $\mathsf{Ve}_{D,x}$ the set of all resulting clauses (after variable elimination of $x$).

There is a technical complication in that ground instances of clauses in $\mathsf{Ve}_{D,x}$ are not necessarily smaller than the corresponding ground instances of $D\sigma$. Fortunately, it can be shown that those ground instances of $D$ in which the unshielded variable $x$ is instantiated to a maximal term in the clause, are indeed $\mathsf{EC_P}$-redundant. This implies that variable chaining is not needed.

LEMMA 18. *Let $D$ be a clause as indicated above, with an unshielded variable $x$, and let $\sigma$ 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 $\mathsf{Ve}_{D,x}$ and the totality axiom.*

To sum up these considerations, clauses with an unshielded variable $x$ are in many, but not all, cases redundant. But inferences involving the unshielded variable (chaining through $x$, superposition from $x$, [in]equality factoring applied to [in]equalities with $x$ as maximal term) are always redundant: the ordering constraints require $x$ to be instantiated with the maximal term, in which case the inference is redundant, as indicated by the preceding lemma. In short, variable chaining can be avoided, whenever the above variable elimination rule is sound. That is the case for dense total orderings without endpoints.

By $\mathcal{CV}^{\succ}$ we denote the calculus $\mathcal{O}^{\succ}$ plus variable elimination, but without inferences in which the maximal term of an instantiated premise is obtained from an unshielded variable.

THEOREM 5. *Let $\succ$ be a reduction ordering that is admissible for variable elimination in inequalities. Let $N$ be a set of clauses that contains the irreflexivity and totality axioms and is saturated up to $\mathsf{EC_P}$-redundancy with respect to $\mathcal{CV}^{\succ}$. Then*

| Problem | Saturate | | | SPASS | | |
|---|---|---|---|---|---|---|
| | derived | kept | time | derived | kept | time |
| Split3 | 170 | 51 | 2.9 | 2372 | 1438 | 1.9 |
| Split8 | 78 | 52 | 1.7 | 111 | 108 | 0.1 |
| Append | 169 | 50 | 2.7 | 16293 | 4907 | 114.9 |
| Permutation | 676 | 255 | 25.3 | — | — | — |
| Ivt | 81 | 43 | 1.5 | — | — | — |
| Cont | 566 | 98 | 51.6 | — | — | — |

Table 1. Experimental Results with Saturate and SPASS

*the set* $N \cup \mathsf{Do}(d, l, r) \cup \mathsf{EC_P}$*, where* $d$*,* $l$ *and* $r$ *are function symbols not occurring in* $N$*, is unsatisfiable if and only if* $N$ *contains the empty clause.*

PROOF. It suffices to show that the set $N \cup \mathsf{Do}(d, l, r)$ is saturated up to $\mathsf{EC_P}$-redundancy. The only possible inferences between clauses in $N$ and $\mathsf{Do}(d, l, r)$ are variable chainings, which have been excluded from $\mathcal{CV}^{\succ}$. The only inference from clauses in $\mathsf{Do}(d, l, r)$ is by ordered chaining with the two clauses containing $d$. This produces a tautology and is therefore redundant too. Since $N$ is assumed to be saturated, all inferences from clauses in $N$ are redundant in $N$ and hence in $N \cup \mathsf{Do}(d, l, r)$. In short, $N \cup \mathsf{Do}(d, l, r)$ is saturated up to $\mathsf{EC_P}$-redundancy. □

The theorem applies whenever the given reduction ordering is admissible for variable elimination. On the other hand, we already pointed out that effective saturation is possible if all (non-redundant) ground inferences are decreasing, which requires strong admissibility in some cases. The following orderings satisfy both requirements. Let $\succ$ be a total well-founded ordering on ground terms. If $L$ is a positive inequality in a ground instance $C\sigma$ of a given clause, we associate with $L$ the quintuple $(max_L, p_L, s_L, a_L, n_L)$; otherwise we associate with $L$ the quintuple $(max_L, p_L, s_L, a_L, d_L)$. These tuples are compared lexicographically as before (with the greater-than relation in the last component). Then we extend the ordering $\succ$ to literals: $L \succ L'$ whenever the tuple associated with $L'$ is strictly smaller than the tuple associated with $L$. Any total well-founded extension of this ordering is admissible for variable elimination in inequalities and strongly admissible for the set of all special predicates other than inequality. Such an extension can be obtained by extending the quintuples by an additional component $min_t$, which contains the non-maximal argument term of a literal, and using a total extension of the (possibly partial) ordering $<_{\mathsf{P}}$ in the third component.

## 7. EXPERIMENTAL RESULTS

Table 1 displays some experimental results that we have obtained with the Saturate[Ganzinger et al. 1994] and SPASS[Weidenbach 1997] theorem provers. The first part of the table refers to lemmas needed to verify quicksort: "Split3" and "Split8" express the main properties of an array after rearranging (splitting) it according to some pivot element; "Append" states that the result of appending two sorted subarrays is sorted; and "Permutation" indicates that the composition of three partial permutations is another partial permutation of an array. These problems involve both the equality predicate and a total ordering. The second part of the table contains information about automatic proofs of the intermediate value theorem and the theorem that the sum of two continuous functions is continuous. These problems involve a dense total ordering without end points.

Saturate is an experimental theorem prover written in Prolog in which the chaining and variable elimination techniques as described in this paper have been implemented. Saturate also contains $\mathsf{EC_P}$-redundancy elimination techniques, such as tautology elimination, recursive conditional reduction for clauses with equalities and/or inequalities, condensement, and subsumption. SPASS represents a very efficient, state-of-the-art implementation of the superposition calculus, plus sort constraints, together with essential simplification techniques. It has no special treatment of transitive relations other than equality, though.

We observe that on the first three problems SPASS derives many more clauses (by up to two orders of magnitude) than Saturate until a proof is found. Also note that the number of clauses that are kept (that is, not eliminated because of the built-in redundancy criteria) is much larger than for Saturate. For the other three examples SPASS did not find a proof within one hour of cpu time, even though it is considerably faster than Saturate, as the first three examples indicate. (All times are in cpu seconds.)

The better performance of Saturate appears to lie mainly in the additional simplification techniques, in particular the variable elimination techniques for total orderings, and the following form of *unordered* subsumption resolution:

$$\frac{C\,,\, u \trianglelefteq_1 v \quad D\,,\, C\sigma\,,\, (v \trianglelefteq_2 u)\sigma}{C\sigma\,,\, D},$$

if at least one symbol $\trianglelefteq_i$ is the strict inequality $<$. This inference is a simplification rule as the conclusion subsumes the second premise. If the main and side literal of this inference are strictly maximal in their clauses, the inference is an instance of ordered (subterm) chaining followed by irreflexivity resolution. But since it is a simplification rule, we perform it even when the ordering constraints for these inferences are not satisfied.

## 8. CONCLUSIONS

We have shown how to extend standard term rewriting techniques to transitive relations other than equality. A key difference between our approach and previous chaining methods is that we impose various ordering constraints on inferences. We have outlined the basic concepts of this approach and have shown how ordering restrictions may be derived from suitable commutation properties. Various inference systems have been presented: a basic calculus $\mathcal{B}$ for transitivity, a generalized calculus $\mathcal{C}$ for collections of composition laws, a subterm chaining calculus $\mathcal{SC}$ for the combination of composition laws and axioms specifying a congruence relation, and finally calculi $\mathcal{O}$ and $\mathcal{CV}$ as further extensions of subterm chaining for dealing with total orderings.

Our results about ordered chaining improve previous work on chaining-based inference systems in several respects. Our calculi apply to sets of special relations related by composition laws, not just to one or two relations, and our completeness results cover first-order languages with arbitrary additional (predicate and function) symbols. We have also integrated the treatment of equality and other transitive relations, so that known results about equational reasoning (e.g., the superposition calculus) are obtained as special cases. Moreover, we have proved that ordered chaining is compatible with a general notion of redundancy whereby common simplification and deletion techniques can be justified by simply showing that deleted clauses or ignored inferences are redundant. In particular, we have demonstrated that variable elimination techniques can often be justified by redundancy arguments, e.g., in the case of total dense orderings without endpoints.

The theory described here has been implemented in the Saturate system with promising initial experiments. Automated theorem provers often become tied up in deducing too many uninteresting and irrelevant consequences of syntactically cumbersome axioms such as the transitivity law or the totality axiom. In other words, the addition of clauses that represent structural knowledge about a problem domain may often increase the search space for deduction to such an extent that an automated proof search becomes unmanagable. In a prover such as Saturate, on the other hand, structural knowledge is exploited by way of specialized inference rules and redundancy elimination techniques designed to speed up the proof search. In such provers additional domain knowledge will often result in qualitative improvements, as generic deductive proof search is replaced by domain-specific (and don't-care non-deterministic) simplification.

The problem of integrating mathematical decision procedures and automated theorem provers is closely related to the replacement of deductive search by suitable simplification techniques, and the issues are also fundamental in the design of heuristic theorem provers, such as the Boyer-More theorem prover or the PVS-System [Owre et al. 1992], that heavily depend on simplification techniques and decision procedures. (Boyer and Moore [1988] provide an interesting discussion about the integration of mathematical theories in a heuristic prover.) The approach taken here, as exemplified by quantifier (i.e., variable) elimination for dense total orderings, aims at an integration that guarantees refutational completeness in the presence of additional predicate and function symbols. One obtains a decision procedure for a certain subtheory, whereas in general deductive search is to a large extent replaced by simplification. The methods developed in this paper represent only a first small step towards the integration of domain-specific knowledge in saturation-based theorem provers. A possible direction for further research is the investigation of an extension of our methods to richer algebraic theories with transitive relations, say ordered groups or fields, or certain fragments of lattice theory.

ACKNOWLEDGMENTS

We would like to thank Jürgen Stuber for his help with running the experiments and the reviewers for their comments on the preliminary version of the paper.

REFERENCES

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

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

Bachmair, L. and Ganzinger, H. 1990. On restrictions of ordered paramodulation with simplification. In M. Stickel Ed., *Proc. 10th Int. Conf. on Automated Deduction, Kaiserslautern*, Volume 449 of *Lecture Notes in Computer Science* (Berlin, 1990), pp. 427–441. Springer-Verlag.

Bachmair, L. and Ganzinger, H. 1994. Rewrite-based equational theorem proving with selection and simplification. *J. Logic and Computation 4*, 3, 217–247. Revised version of Technical Report MPI-I-91-208, 1991.

Bachmair, L. and Ganzinger, H. 1995. Ordered chaining calculi for first-order theories of binary relations. Research Report MPI-I-95-2-009, Max-Planck-Institut für Informatik, Saarbrücken, Saarbrücken.

Bachmair, L. and Ganzinger, H. 1997. A theory of resolution. Research Report MPI-I-97-2-005, Max-Planck-Institut für Informatik, Saarbrücken, Saarbrücken. To appear in the Handbook of Automated Reasoning.

Bachmair, L., Ganzinger, H., and Stuber, J. 1995. Combining algebra and universal algebra in first-order theorem proving: The case of commutative rings. In E. Astesiano, G. Reggio, and A. Tarlecki Eds., *Recent Trends in Data Type Specification*, Volume 906 of *Lecture Notes in Computer Science* (Berlin, 1995), pp. 1–29. Springer-Verlag.

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

Boyer, R. S. and Moore, J. S. 1988. Integrating decision procedures into heuristic theorem provers: A case study of linear arithmetic. In J. E. Hayes, D. Michie, and J. Richards Eds., *Machine Intelligence 11*, Chapter 5, pp. 83–124. Oxford: Clarendon Press.

Dershowitz, N. 1987. Termination of rewriting. *J. Symbolic Computation 3*, 69–116.

Dershowitz, N. 1991. A maximal-literal unit strategy for horn clauses. In *Proc. 2nd Int. Workshop on Conditional and Typed Rewriting Systems*, Volume 516 of *Lecture Notes in Computer Science* (Berlin, 1991), pp. 14–25. Springer-Verlag.

Ganzinger, H. 1991. A completion procedure for conditional equations. *J. Symbolic Computation 11*, 51–81.

Ganzinger, H., Nieuwenhuis, R., and Nivela, P. 1994. The Saturate system. User manual maintained on the Web.

Hines, L. M. 1992. Completeness of a prover for dense linear logics. *J. Automated Reasoning 8*, 45–75.

Hines, L. M. 1994. Str+ve and integers. In A. Bundy Ed., *Twelfth International Conference on Automated Deduction*, LNAI 814 (Nancy, France, June 26– July 1 1994), pp. 416–430. Springer-Verlag.

Levy, J. and Agustí, J. 1993. Bi-rewriting, a term rewriting technique for monotonic order relations. In C. Kirchner Ed., *Rewriting Techniques and Applications*, Volume 690 of *Lecture Notes in Computer Science* (Berlin, 1993), pp. 17–31. Springer.

Manna, Z. and Waldinger, R. 1986. Special relations in automated deduction. *J. Association for Computing Machinery 33*, 1–60.

Manna, Z. and Waldinger, R. 1992. The special-relation rules are incomplete. In D. Kapur Ed., *11th International Conference on Automated Deduction* (Saratoga Springs, NY, June 1992), pp. 492–506. Springer LNAI 607.

Owre, S., Rushby, J. M., and Shankar, N. 1992. PVS: A prototype verification system. In *11th Int. Conf. on Automated Deduction*, Volume 607 of *LNCS* (1992), pp. 748–752. Springer.

Richter, M. 1984. Some reordering properties for inequality proof trees. Volume 171 of *Lecture Notes in Computer Science* (1984), pp. 183–197. Springer-Verlag.

Scott, D. S. 1979. Identity and existence in intuitionistic logic. In *Applications of Sheaves, Proc. Durham*, Volume 753 of *Lecture Notes in Mathematics*, pp. 660–695.

Slagle, J. R. 1972. Automatic theorem proving with built-in theories including equality, partial orderings, and sets. *J. Association for Computing Machinery 19*, 120–135.

Waldmann, U. 1998. Superposition for divisible torsion-free abelian groups. In C. Kirchner and H. Kirchner Eds., *Automated Deduction – CADE-15, 15th International Conference on Automated Deduction*, LNAI 1421, pp. 144–159. Springer-Verlag.

WEIDENBACH, C. 1997. Spass version 0.49. *J. Automated Reasoning 18*, 2, 247–252.