Rewrite-Based Equational Theorem Proving With Selection and Simplification

Leo Bachmair
Harald Ganzinger

MPI–I–91–208 September 1991

Authors' Addresses

Leo Bachmair, Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794, U.S.A, `leo@sbcs.sunysb.edu`
Harald Ganzinger, Max-Planck-Institut für Informatik, Im Stadtwald, D-W-6600 Saarbrücken, Germany, `hg@cs.uni-sb.de`

Publication Notes

An extended abstract of this paper appeared in the proceedings of the *Second Int. Workshop on Conditional and Typed Rewriting*, Montréal 1990. The present report has been submitted for publication elsewhere and will be copyrighted if accepted.

Acknowledgements

Marianne Haberstrau and Robert Nieuwenhuis have found two errors in a preliminary version of the paper. We are very grateful for their careful reading. We also wish to thank Uwe Waldmann for his valuable comments.

The research described in this paper was supported in part by the National Science Foundation under grant CCR-8901322, by the ESPRIT project PROSPECTRA (ref. no. 390), and by a travel grant from the Deutsche Forschungsgemeinschaft.

Abstract

We present various refutationally complete calculi for first-order clauses with equality that allow for arbitrary selection of negative atoms in clauses. Refutation completeness is established via the use of well-founded orderings on clauses for defining a Herbrand model for a consistent set of clauses. We also formulate an abstract notion of redundancy and show that the deletion of redundant clauses during the theorem proving process preserves refutation completeness. It is often possible to compute the closure of nontrivial sets of clauses under application of non-redundant inferences. The refutation of goals for such complete sets of clauses is simpler than for arbitrary sets of clauses, in particular one can restrict attention to proofs that have support from the goals without compromising refutation completeness. Additional syntactic properties allow to restrict the search space even further, as we demonstrate for so-called quasi-Horn clauses. The results in this paper contain as special cases or generalize many known results about Knuth-Bendix-like completion procedures (for equations, Horn clauses, and Horn clauses over built-in Booleans), completion of first-order clauses by clausal rewriting, and inductive theorem proving for Horn clauses.

Keywords

Theorem Proving, First-Order Logic, Equality, Paramodulation, Rewrite Techniques, Simplification, Saturation

1 Introduction

Methods for dealing with the equality predicate are of central concern in automated theorem proving. One of the more successful approaches to equational theorem proving is based on the use of equations as (one-way) rewrite rules. For instance, the so-called completion method (Knuth and Bendix 1970) attempts to construct a convergent (i.e., terminating and Church-Rosser) rewrite system for a given set of (universally quantified) equational axioms. Two terms can be rewritten to identical normal forms if, and only if, they are equal. A convergent rewrite system thus provides a decision procedure for its equational theory. The completion procedure may fail in general, but has been extended to a refutationally complete theorem prover (cf. Lankford 1975, Hsiang and Rusinowitch 1987, and Bachmair, Dershowitz and Plaisted 1989). Completion procedures for conditional equations (i.e., Horn clauses with equations as the only atomic formulas) have been described by Kounalis and Rusinowitch (1988), and by Ganzinger (1987a, b).

The two main components of completion are (i) the deductive inference rule of *superposition* and (ii) various mechanisms for deleting redundant equations via *simplification by rewriting*. There have been several attempts to extend completion to first-order clauses, based on the observation that superposition is a restricted form of paramodulation (Robinson and Wos 1969). Another technique common in clausal theorem proving, demodulation (Wos et al. 1967), is essentially a special case of simplification by rewriting.

Consider, for instance, paramodulation (for variable-free formulas):

$$\frac{\Gamma \rightarrow \Delta, s \approx t \quad \Lambda \rightarrow \Pi, u[s] \approx v}{\Gamma, \Lambda \rightarrow \Delta, \Pi, u[t] \approx v}$$

and suppose that $\succ$ is an ordering which is total on variable-free terms and formulas. We say that the paramodulation inference is *ordered* (with respect to $\succ$) if (i) $s \succ t$; (ii) $s \approx t$ is strictly maximal with respect to $\Gamma \cup \Delta$; and (iii) $u[s] \approx v$ is strictly maximal with respect to $\Lambda \cup \Pi$. An ordered paramodulation inference is said to be a *superposition inference* if (iv) $u[s] \succ v$. The superposition is called *strict* if in addition (v) s does not occur in Γ. A *weak superposition* inference is a paramodulation inference for which conditions (i), (iii), and (iv)—but not necessarily (ii)—are satisfied.

Hsiang and Rusinowitch (1989) have proved that ordered paramodulation is refutationally complete, whereas Rusinowitch (1991) has established the refutation completeness of weak superposition. Strict superposition is unfortunately not complete (Bachmair and Ganzinger 1990).

For example, consider the set of clauses

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

where a, b, c, and d are constants. This set is unsatisfiable: from the last three clauses we may infer that $a \approx b \approx c \approx d$, which contradicts the first clause. However, if $\succ$ is an ordering in which $a \succ b \succ c \succ d$, then the only clause that can be obtained from the above clauses by superposition is $a \approx d \rightarrow b \approx c, a \approx d$, which is a tautology. No further clauses, in particular no contradiction, can be deduced by superposition.

However, several moderate enrichments of the (strict) superposition calculus are indeed refutationally complete (Bachmair and Ganzinger 1990). In this paper we generalize the deductive part of (conditional) completion to first-order clauses, and more importantly also deal with simplification, which is indispensable for any practical use of completion.[1] We introduce an abstract notion of redundancy (of clauses and inferences) and use it as the fundamental concept in formulating a framework for theorem proving with simplification. We present criteria that can be used in tests for checking redundancy and show that under reasonable conditions on the search strategy employed by a theorem prover, deletion of redundant clauses does not destroy refutation completeness. The notion of saturation of a set of clauses, in the sense that all non-redundant inferences are computed, generalizes completion (with simplification) to first-order clauses. In addition, we show that arbitrary selection functions on negative literals can be used with these superposition calculi.

The simplification techniques to which our results apply include, among others, deletion of tautologies, subsumption, case analysis, and contextual reductive rewriting. We also investigate ways of improving the search for a refutation of a given goal with respect to a set of clauses that is already saturated. The results presented here for the case of first-order clauses include and generalize results about ordered completion of equations (Bachmair, Dershowitz and Plaisted 1989; Bachmair 1991), completion of Horn clauses (Kounalis and Rusinowitch 1988, Ganzinger 1987b), and ground completion of Horn clauses over built-in Booleans (Zhang and Rémy 1985, Ganzinger 1987a, Nieuwenhuis and Orejas 1991).

The paper is organized as follows. In the next chapter we introduce our terminology and basic definitions. We describe superposition calculi with selection functions on negative literals in Chapter 3 and prove their refutation completeness in Chapter 4. The notions of redundancy and saturation, as well as modular criteria for redundancy, are also introduced in Chapter 4. In Chapter 5 we outline an abstract framework for theorem proving with simplification and discuss various specific simplification mechanisms, such as case analysis and contextual reductive rewriting. In Chapter 6 we study refutation of goals for so-called quasi-Horn programs and include various results about Horn clauses and inductive theorem proving.

2 Preliminaries

2.1 Equational clauses

We formulate our inference rules in an equational framework and define clauses in terms of multisets.

A *multiset* over a set X is a function M from X to the natural numbers. Intuitively, $M(x)$ specifies the number of occurrences of x in M. We say that x is an *element* of M if $M(x) > 0$, and M is a *submultiset* of M' (written $M \subseteq M'$) if $M(x) \leq M'(x)$, for all x. A multiset M is called *finite* if $M(x) = 0$ for all but finitely many x. The *union* and *intersection* of multisets are defined by the identities $M_1 \cup M_2(x) = M_1(x) + M_2(x)$ and $M_1 \cap M_2(x) = \min(M_1(x), M_2(x))$. If M is a multiset and S a set, we write $M \subseteq S$ to indicate that every element of (the multiset) M is an element of (the set) S, and use $M \cap S$ to denote the *set* $\{x \in S : M(x) \geq 1\}$. For simplicity, we

[1]Rusinowitch (1991) does discuss simplification to some extent, but for practical purposes his simplification techniques are inadequate even for the very simplest case—completion of sets of universally quantified equations.

often use a set-like notation to describe multisets. For example, $\{x, x, x\}$ denotes the multiset M for which $M(x) = 3$ and $M(y) = 0$, for $y \neq x$.

An *equation* is an expression $s \approx t$, where s and t are (first-order) terms built from given function symbols and variables. We identify $s \approx t$ with the multiset $\{s, t\}$. By a *ground* expression (i.e., a term, equation, formula, etc.) we mean an expression containing no variables.

A *clause* is a pair of multisets of equations, written $\Gamma \to \Delta$. The multiset Γ is called the *antecedent*; the multiset Δ, the *succedent*. We usually write Γ_1, Γ_2 instead of $\Gamma_1 \cup \Gamma_2$; Γ, A or A, Γ instead of $\Gamma \cup \{A\}$; and $A_1, \ldots, A_m \to B_1, \ldots, B_n$ instead of $\{A_1, \ldots, A_m\} \to \{B_1, \ldots, B_n\}$. A clause $A_1, \ldots, A_m \to B_1, \ldots, B_n$ represents an implication $A_1 \wedge \cdots \wedge A_m \supset B_1 \vee \cdots \vee B_m$; the empty clause, a contradiction. Clauses of the form $\Gamma, A \to A, \Delta$ or $\Gamma \to \Delta, t \approx t$ are called *tautologies*.

2.2 Equality Herbrand interpretations

We write $A[s]$ to indicate that A contains s as a subexpression and (ambiguously) denote by $A[t]$ the result of replacing a particular occurrence of s by t. By $A\sigma$ we denote the result of applying the substitution σ to A and call $A\sigma$ an *instance* of t. If $A\sigma$ is ground, we speak of a *ground instance*. We shall also consider instances of multisets of equations and of clauses. For example, the multiset $\{a \approx b, a \approx b\}$ is an instance of $\{x \approx b, a \approx y\}$. Composition of substitutions is denoted by juxtaposition. Thus, if τ and ρ are substitutions, then $x\tau\rho = (x\tau)\rho$, for all variables x.

An *equivalence* is a reflexive, transitive, symmetric binary relation. An equivalence $\sim$ on terms is called a *congruence* if $s \sim t$ implies $u[s] \sim u[t]$, for all terms u, s, and t. If E is a set of ground equations, we denote by E^* the smallest congruence $\sim$ such that $s \sim t$ whenever $s \approx t \in E$.

By an (*equality Herbrand*) *interpretation* we mean a congruence on ground terms. An interpretation I is said to *satisfy* a ground clause $\Gamma \to \Delta$ if either $\Gamma \not\subseteq I$ or else $\Delta \cap I \neq \emptyset$. We also say that a ground clause C is *true in* I, if I satisfies C; and that C is *false in* I, otherwise. An interpretation I is said to satisfy a non-ground clause $\Gamma \to \Delta$ if it satisfies all ground instances $\Gamma\sigma \to \Delta\sigma$. For instance, a tautology is satisfied by any interpretation. A clause which is satisfied by no interpretation (e.g., the empty clause) is called *unsatisfiable*. An interpretation I is called a (*equality Herbrand*) *model* of N if it satisfies all clauses of N. A set N of clauses is called *consistent* if it has a model; and *inconsistent* (or *unsatisfiable*), otherwise. We say that N *implies* C, and write $N \models C$, if every model of N satisfies C.

Convergent rewrite systems provide a convenient formalism for describing and reasoning about equality interpretations.

2.3 Convergent rewrite systems

A binary relation $\Rightarrow$ on terms is called a *rewrite relation* if $s \Rightarrow t$ implies $u[s\sigma] \Rightarrow u[t\sigma]$, for all terms s, t and u, and substitutions σ. A transitive, well-founded rewrite relation is called a *reduction ordering*. By $\Leftrightarrow$ we denote the symmetric closure of $\Rightarrow$; by $\Rightarrow^*$ the transitive, reflexive closure; and by $\Leftrightarrow^*$ the symmetric, transitive, reflexive closure. Furthermore, we write $s \Downarrow t$ to indicate that s and t can be rewritten to a common form: $s \Rightarrow^* v$ and $t \Rightarrow^* v$, for some term v. A rewrite relation $\Rightarrow$ is said to be *Church-Rosser* if the two relations $\Leftrightarrow^*$ and $\Downarrow$ are the same.

A set of equations E is called a *rewrite system* with respect to an ordering $\succ$ if we have $s \succ t$ or $t \succ s$, for all equations $s \approx t$ in E. If all equations in E are ground, we speak of a ground rewrite system. Equations in E are also called (*rewrite*) *rules*. When we speak of "the rule $s \approx t$"

we implicitly assume that $s \succ t$. By $\Rightarrow_{E^\succ}$ (or simply $\Rightarrow_E$) we denote the smallest rewrite relation for which $s \Rightarrow_E t$ whenever $s \approx t \in E$ and $s \succ t$. A term s is said to be in *normal form* (with respect to E) if it can not be rewritten by $\Rightarrow_E$, i.e., if there is no term t such that $s \Rightarrow_E t$. A term is also called *irreducible*, if it is in normal form, and *reducible*, otherwise. For instance, if $s \Downarrow_E t$ and $s \succ t$, then s is reducible by E.

A rewrite system E is said to be *convergent* if the rewrite relation $\Rightarrow_E$ is well-founded and Church-Rosser. Convergent rewrite systems define unique normal forms. A ground rewrite system E is called *left-reduced* if for every rule $s \approx t$ in E the term s is irreducible by $E \setminus \{s \approx t\}$. It is well-known that left-reduced, well-founded rewrite systems are convergent (cf. Huet 1980).

2.4 Clause orderings

Any ordering $\succ$ on a set S can be extended to an ordering $\succ_{mul}$ on finite multisets over S as follows: $M \succ_{mul} N$ if (i) $M \neq N$ and (ii) whenever $N(x) > M(x)$ then $M(y) > N(y)$, for some y such that $y \succ x$. If $\succ$ is a total [well-founded] ordering, so is $\succ_{mul}$. Given a set (or multiset) S and an ordering $\succ$ on S, we say that x is *maximal* relative to S if there is no $y \in S$ with $y \succ x$; and *strictly maximal* if there is no $y \in S$ with $y \succeq x$.

If $\succ$ is an ordering on terms, then the corresponding multiset ordering $\succ_{mul}$ is an ordering on equations, which we denote by $\succ^e$.

We have defined clauses as pairs of multisets of equations. Alternatively, clauses may also be thought of as multisets of *occurrences* of equations. We identify an occurrence of an equation $s \approx t$ in the antecedent of a clause with the multiset (of multisets) $\{\{s, \bot\}, \{t, \bot\}\}$, and an occurrence in the succedent with the multiset $\{\{s\}, \{t\}\}$, where $\bot$ is a new symbol.[2] We identify clauses with finite multisets of occurrences of equations. By $\succ^o$ we denote the twofold multiset ordering $(\succ_{mul})_{mul}$ of *succ*, which is an ordering on occurrences of equations; by $\succ^c$ we denote the multiset ordering $\succ^o_{mul}$, which is an ordering on clauses. If $\succ$ is a well-founded [total] ordering, so are $\succ^e$, $\succ^o$, and $\succ^c$.

Observe the difference between the ordering $\succ^e$ on equations and the ordering $\succ^o$ on occurrences of equations. For example, if $s \succ t \succ u$, then $s \approx t \succ^e s \approx u$, but nonetheless we have $\Gamma, s \approx u \rightarrow \Delta \succ^c \Gamma \rightarrow s \approx t, \Delta$ as the occurrence of $s \approx u$ (in the antecedent) is larger than the occurrence of $s \approx t$ (in the succedent).

The superposition calculi described below are defined in terms of these orderings.[3] In Bachmair and Ganzinger (1990) we have identified an occurrence of an equation $s \approx t$ in the antecedent of a clause with the multiset $\{\{s, t, \bot\}\}$ and an occurrence in the succedent with the multiset $\{\{s, t\}\}$, and consequently obtained a slightly different clause ordering and inference system. The above ordering has the advantage that it eliminates certain technical complications in the proof of the refutation completeness of the corresponding superposition calculus.

[2]The symbol $\bot$ is not part of the vocabulary of the given first-order language. It is assumed to be minimal with respect to any given ordering. Thus $t \succ \bot$, for all terms t.

[3]We shall also use orderings to define simplification techniques that are compatible with superposition. In a few cases the simplification techniques used in completion require information about the substitution by which an instance of a clause is obtained. For that purpose it is necessary to consider pairs (C, σ) of clauses and substitutions, and not just instances $C\sigma$, so that an ordering may distinguish between pairs (C, σ) and (D, τ) even when the instances $C\sigma$ and $D\tau$ are identical. For the sake of simplicity we have chosen not to use this slightly more sophisticated formalism in this paper.

3 Ordered Inference Rules with Selection

We shall consider inference rules

$$\frac{C_1 \cdots C_n}{C}$$

where $C_1, \ldots, C_n$ (the *premises*) and C (the *conclusion*) are clauses.

Definition 1 Let $\succ$ be a reduction ordering. We say that a clause $C = \Gamma \rightarrow \Delta, s \approx t$ is *reductive* for $s \approx t$ if $t \not\succeq s$ and $s \approx t$ is a strictly maximal occurrence of an equation in C.[4]

For example, if $s \succ t \succ u$ and $s \succ v$, for every term v occurring in Γ, then $\Gamma \rightarrow s \approx u, s \approx t$ is reductive for $s \approx t$, but $\Gamma, s \approx u \rightarrow s \approx t$ is not. In general, if a clause C is reductive for $s \approx t$, then s must not occur in the antecedent of C.

The following inference rules are defined with respect to $\succ$. The first rule encodes the reflexivity of equality:

Equality resolution: $$\frac{\Lambda, u \approx v \rightarrow \Pi}{\Lambda\sigma \rightarrow \Pi\sigma}$$

where σ is a most general unifier of u and v and $u\sigma \approx v\sigma$ is a maximal occurrence of an equation in $\Lambda\sigma, u\sigma \approx v\sigma \rightarrow \Pi\sigma$.

The next inference rule represents a variant of factoring, restricted to the succedent of clauses:

Ordered factoring: $$\frac{\Gamma \rightarrow \Delta, A, B}{\Gamma\sigma \rightarrow \Delta\sigma, A\sigma}$$

where σ is a most general unifier of A and B and $A\sigma$ is a maximal occurrence of an equation in $\Gamma\sigma \rightarrow \Delta\sigma, A\sigma, B\sigma$.

The following superposition rules represent restricted versions of paramodulation:

Superposition, left: $$\frac{\Gamma \rightarrow \Delta, s \approx t \quad u[s'] \approx v, \Lambda \rightarrow \Pi}{u[t]\sigma \approx v\sigma, \Gamma\sigma, \Lambda\sigma \rightarrow \Delta\sigma, \Pi\sigma}$$

where (i) σ is a most general unifier of s and s', (ii) the clause $\Gamma\sigma \rightarrow \Delta\sigma, s\sigma \approx t\sigma$ is reductive for $s\sigma \approx t\sigma$, (iii) $v\sigma \not\succeq u\sigma$ and $u\sigma \approx v\sigma$ is a maximal occurrence of an equation in $u\sigma \approx v\sigma, \Lambda\sigma \rightarrow \Pi\sigma$,[5] and (iv) s' is not a variable.

Superposition, right: $$\frac{\Gamma \rightarrow \Delta, s \approx t \quad \Lambda \rightarrow u[s'] \approx v, \Pi}{\Gamma\sigma, \Lambda\sigma \rightarrow u[t]\sigma \approx v\sigma, \Delta\sigma, \Pi\sigma}$$

where (i) σ is a most general unifier of s and s', (ii) the clause $\Gamma\sigma \rightarrow \Delta\sigma, s\sigma \approx t\sigma$ is reductive for $s\sigma \approx t\sigma$, (iii) the clause $\Lambda\sigma \rightarrow u\sigma \approx v\sigma, \Pi\sigma$ is reductive for $u\sigma \approx v\sigma$, and (iv) s' is not a variable.

[4]If the reduction ordering $\succ$ is assumed to be *complete* (i.e., total on ground terms), we could use a somewhat stronger formulation of reductivity where instead of $t \not\succeq s$ we require that $s\sigma \succ t\sigma$, for some ground substitution σ. However, our results apply not only to complete orderings, but more generally to *completable* orderings (i.e., orderings contained in a complete ordering); cf. the discussion of ordered completion in Bachmair (1991).

[5]Since we do not require factoring in the antecedent, the equation $u\sigma \approx v\sigma$ may also occur in $\Lambda\sigma$.

The example given in the introduction shows that superposition is not refutationally complete, but has to be combined with additional inference rules. For instance, we may add the following inference rule, which in essence generalizes ordered factoring:

Equality factoring: $$\frac{\Gamma \to \Delta, s \approx t, s' \approx t'}{\Gamma\sigma, t\sigma \approx t'\sigma \to \Delta\sigma, s'\sigma \approx t'\sigma}$$

where (i) σ is a most general unifier of s and s'; (ii) $t\sigma \not\succ s\sigma$ and $t'\sigma \not\succ s'\sigma$; and (iii) $s\sigma \approx t\sigma$ is a maximal occurrence of an equation in $\Gamma\sigma \to \Delta\sigma, s\sigma \approx t\sigma, s'\sigma \approx t'\sigma$.

An alternative to equality factoring is the paramodulation rule:

Merging Paramodulation: $$\frac{\Gamma \to \Delta, s \approx t \quad \Lambda \to u \approx v[s'], u' \approx v', \Pi}{\Gamma\sigma, \Lambda\sigma \to u\sigma \approx v[t]\sigma, u\sigma \approx v'\sigma, \Delta\sigma, \Pi\sigma}$$

where (i) σ is the composition $\tau\rho$ of a most general unifier τ of s and s', and a most general unifier ρ of $u\tau$ and $u'\tau$, (ii) the clause $\Gamma\sigma \to \Delta\sigma, s\sigma \approx t\sigma$ is reductive for $s\sigma \approx t\sigma$, (iii) the clause $\Lambda\sigma \to \Pi\sigma, u\sigma \approx v\sigma, u'\sigma \approx v'\sigma$ is reductive for $u\sigma \approx v\sigma$, (iv) $u\tau \succ v\tau$ and $v'\sigma \not\succeq v\sigma$, and (v) s' is not a variable.

Merging paramodulation is designed in such a way that its repeated application to ground clauses (in conjunction with ordered factoring) has the effect of merging atoms in the succedent containing a maximal term.

By $\mathcal{E}$ we denote the inference system consisting of equality resolution, equality factoring, and superposition; by $\mathcal{P}$ the inference system consisting of equality resolution, ordered factoring, superposition, and merging paramodulation. (If necessary, we indicate the underlying ordering by writing $\mathcal{E}^{\succ}$ or $\mathcal{P}^{\succ}$.) In each case the following additional restrictions are imposed: (a) the premises of an inference rule must not share any variables (if necessary, the variables in one premise are renamed); and (b) if C and D are the premises of a paramodulation inference with σ the mgu obtained from superposing C on D, then $C\sigma \not\succeq^c D\sigma$. Later on we will define a notion of redundancy for inferences which restricts the inference systems even further. For example, inferences involving tautologies will be redundant.

An essential property of the above inference rules is that the conclusion of a ground inference is always simpler (with respect to the ordering $\succ^c$) than the maximal premise (which is always the second premise in the case of a paramodulation inference).

We shall also discuss variants of the above inference rules that are controlled by a *selection function* that assigns to each clause a (possibly empty) multiset of (occurrences of) equations in the antecedent. If S is such a selection function, then the equations in $S(C)$ are called *selected*. Selected equations can be arbitrarily chosen and need not be maximal.[6] Also, $S(C) = \emptyset$ indicates that no equation is selected.

The following inference rules are defined with respect to a given selection function S.

Selective resolution: $$\frac{\Lambda, u \approx v \to \Pi}{\Lambda\sigma \to \Pi\sigma}$$

where σ is a most general unifier of u and v and $u \approx v$ is a selected equation in $\Lambda, u \approx v \to \Pi$.

[6]An ordering may be used, however, for further distinctions if more than one equation is selected. The corresponding modifications required for this technique are straightforward and will not be discussed below.

$$\textbf{Selective superposition}: \quad \frac{\Gamma \to \Delta, s \approx t \quad u[s'] \approx v, \Lambda \to \Pi}{u[t]\sigma \approx v\sigma, \Gamma\sigma, \Lambda\sigma \to \Delta\sigma, \Pi\sigma}$$

where (i) σ is a most general unifier of s and s', (ii) the clause $C = \Gamma \to \Delta, s \approx t$ contains no selected equations and $C\sigma$ is reductive for $s\sigma \approx t\sigma$, (iii) $v\sigma \not\succeq u\sigma$ and $u \approx v$ is a selected equation in $u \approx v, \Lambda \to \Pi$, and (iv) s' is not a variable.

By $\mathcal{E}_S$ [$\mathcal{P}_S$] we denote the system consisting of the above two selective inference rules plus all inference rules of $\mathcal{E}$ [$\mathcal{P}$], with the additional restriction on the latter rules that no premise contain any selected literals. For instance, in the presence of a selection function right superposition can only be applied to premises that contain no selected equations.

A selection function S for which $S(C)$ is a singleton whenever the antecedent of C is non-empty, is called *unitary*. An inference system for Horn clauses with arbitrary unitary selection functions, which is based on the inference system $\mathcal{E}_S$, has been described by Ganzinger (1987a). (The two inference systems $\mathcal{E}_S$ and $\mathcal{P}_S$ are identical in the Horn clause case, as ordered factoring, equality factoring, and merging paramodulation cannot be applied to Horn clauses.)

A selection function S for which $S(C)$ is non-empty whenever the antecedent of C is non-empty determines a so-called *positive superposition strategy*. In positive strategies only clauses with empty antecedent are paramodulated into other clauses. An example of such an inference system, also for Horn clauses, is the *maximal-literal unit strategy* (Dershowitz 1991), where for each clause the complete antecedent is selected and ordering constraints are imposed on selected equations.

The case of first-order clauses with equality and additional arbitrary predicates is included in the above framework. From now on we shall assume that a set of predicate symbols is given in addition to the set of function symbols. Thus we also consider expressions $P(t_1, \ldots, t_n)$, where P is some predicate symbol and $t_1, \ldots, t_n$ are terms built from function symbols and variables. We then have equations $s \approx t$ between (non-predicate) terms, called *function equations*, and equations $P(t_1, \ldots, t_n) \approx \mathrm{tt}$, called *predicate equations*, where tt is a distinguished unary predicate symbol that is taken to be minimal in the given reduction ordering $\succ$. For simplicity, we usually abbreviate $P(t_1, \ldots, t_n) \approx \mathrm{tt}$ by $P(t_1, \ldots, t_n)$.

Clauses of the form $\Gamma, P(t_1, \ldots, t_n) \approx \mathrm{tt} \to \Delta$ or $\Gamma \to \Delta, P(t_1, \ldots, t_n) \approx \mathrm{tt}$ in which $P(t_1, \ldots, t_n) \approx \mathrm{tt}$ is a maximal equation can evidently not be part of an equality or selective resolution inference. Furthermore, superposition of a clause $\Gamma \to \Delta, P(s_1, \ldots, s_n) \approx \mathrm{tt}$ on a clause $\Lambda \to \Pi, P(t_1, \ldots, t_n) \approx \mathrm{tt}$ results in a tautology $\Gamma\sigma, \Lambda\sigma \to \Delta\sigma, \Pi\sigma, \mathrm{tt} \approx \mathrm{tt}$ and is therefore redundant. The remaining inferences are left or selective superpositions of the form

$$\frac{\Gamma \to \Delta, P(s_1, \ldots, s_n) \approx \mathrm{tt} \quad P(t_1, \ldots, t_n) \approx \mathrm{tt}, \Lambda \to \Pi}{\Gamma\sigma, \Lambda\sigma, \mathrm{tt} \approx \mathrm{tt} \to \Delta\sigma, \Pi\sigma.}$$

Note that the trivial equation $\mathrm{tt} \approx \mathrm{tt}$ in the antecedent can be eliminated by resolution. Thus we obtain a derived inference rule:

$$\textbf{Ordered resolution}: \quad \frac{\Gamma \to \Delta, P(s_1, \ldots, s_n) \quad P(t_1, \ldots, t_n), \Lambda \to \Pi}{\Gamma\sigma, \Lambda\sigma \to \Delta\sigma, \Pi\sigma}$$

with the restrictions associated with selective or left superposition.

4 Refutation Completeness

We shall next prove that the above superposition calculi are refutationally complete in the sense that a contradiction (the empty clause) can be derived from any inconsistent set of clauses. In the proof we shall have to argue about ground instances of given clauses and inferences.

Definition 2 Let π be an inference in $\mathcal{I}$ with premises $C_1, \ldots, C_n$ and conclusion C, where the clauses $C_1, \ldots, C_n$ have no variables in common. By an *instance* of π we mean any inference in $\mathcal{I}$ with premises $C_1\sigma, \ldots, C_n\sigma$ and conclusion $C\sigma$.

It can easily be seen that any resolution or factoring inference from ground instances of N is a ground instance of an inference from N. For superposition this correspondance need not necessarily hold. For instance, let N be the set of three clauses

$$\{\rightarrow a \approx b,\ \rightarrow f(a) \approx f(b),\ p(g(x)) \rightarrow p(f(x))\}.$$

If $a \succ b$, $f(a) \succ f(b)$, and $p(f(x)) \succ p(g(x))$, then

$$\pi = \frac{\rightarrow f(a) \approx f(b) \quad p(g(x)) \rightarrow p(f(x))}{p(g(a)) \rightarrow p(f(b))}$$

is a superposition inference and

$$\frac{\rightarrow f(a) \approx f(b) \quad p(g(a)) \rightarrow p(f(a))}{p(g(a)) \rightarrow p(f(b))}$$

is a ground instance of π, while the superposition inference

$$\frac{\rightarrow a \approx b \quad p(g(a)) \rightarrow p(f(a))}{p(g(a)) \rightarrow p(f(b))}$$

is an inference from ground instances of N, but not a ground instance of any inference from N. Such problematic inferences, which arise from superpositions into the "variable" or "substitution part" of a clause, cannot be "lifted" to the general level, but as we shall see need not be considered.

Definition 3 We say that an instance $C\sigma$ of a clause C is *reduced* with respect to a rewrite system R if $x\sigma$ is irreducible by R, for all variables x occurring in C.

Lemma 1 *Let C and D be clauses with no variables in common, and let $C\sigma = \Gamma \rightarrow \Delta, s \approx t$ and $D\sigma$ be ground instances, such that $D\sigma \succ^c C\sigma$ and $s \approx t$ is a maximal occurrence of an equation in $C\sigma$, and $D\sigma$ is a reduced ground instance of D with respect to $\{s \approx t\}$. Then any superposition or merging paramodulation inference with premises $C\sigma$ and $D\sigma$ is a ground instance of a similar inference from C and D.*

4.1 Construction of Equality Interpretations

Let N be a set of clauses and $\succ$ be a reduction ordering which is total on ground terms. We define an interpretation I by means of a convergent rewrite system R as follows.

First, we use induction on the clause ordering $\succ^c$ to define sets of equations E_C, R_C, and I_C, for all ground instances C of clauses of N. Let C be such a ground instance and suppose that $E_{C'}$, $R_{C'}$, and $I_{C'}$ have been defined for all ground instances C' of N for which $C \succ^c C'$. Then

$$R_C = \bigcup_{C \succ^c C'} E_{C'} \quad \text{and} \quad I_C = R_C^*.$$

Moreover

$$E_C = \{s \approx t\}$$

if C is a clause $\Gamma \rightarrow s \approx t, \Delta$ such that (i) C is reductive for $s \approx t$, (ii) s is irreducible by R_C, (iii) $\Gamma \subseteq I_C$, and (iv) $\Delta \cap I_C = \emptyset$. In that case, we also say that C *produces* the equation (or rule) $s \approx t$. In all other cases, $E_C = \emptyset$. Finally, we define I to be the equality interpretation R^*, where $R = \bigcup_C E_C$ is the set of all equations produced by ground instances of clauses of N.

Clauses that produce equations are also called *productive*. Note that a productive clause C is false in $I_C = R_C^*$, but true in $(R_C \cup E_C)^*$. The sets R_C and R are constructed in such a way that they are left-reduced rewrite systems with respect to $\succ$. Consequently these rewrite systems are convergent and the truth value of an equation can be determined by rewriting: $u \approx v \in I$ if and only if $u \Downarrow_R v$. In many cases the truth value of an equation can already be determined by rewriting with R_C.

Lemma 2 *Let $C = \Gamma \rightarrow \Delta, s \approx t$ be a clause where $s \approx t$ is a maximal occurrence of an equation, and let D be another clause containing s. If $C \succ^c D$ and s is irreducible by R_C, then $R_C = R_D$ (and hence $I_C = I_D$).*

Proof. If C' is any clause with $C \succ^c C' \succeq^c D$, then $E_{C'} = \emptyset$, for otherwise s would be reducible by R_C. Therefore $R_C = R_D \cup \bigcup_{C \succ^c C' \succ^c D} E_{C'} = R_D$. □

Lemma 3 *Let $C = \Gamma, u \approx v \rightarrow \Delta$ and D be ground instances of N with $D \succeq^c C$. Then $u \approx v$ is true in I_C if and only if it is true in I_D if and only if it is true in I.*

Proof. If $u \approx v$ is true in I_C, then $u \Downarrow_{R_C} v$. Since $R_C \subseteq R_D \subseteq R$, we then have $u \Downarrow_{R_D} v$ and $u \Downarrow_R v$, which indicates that $u \approx v$ is true in I_D and in I.

On the other hand, suppose $u \approx v$ is false in I_C. If u' and v' are the normal forms of u and v with respect to R_C, then $u' \neq v'$. Furthermore, if $s \approx t$ is a rule in $R \setminus R_C$, then $s \succ u \succeq u'$ and $s \succ v \succeq v'$. (Clauses which produce rules for terms not greater than u or v are smaller than C.) Therefore, u' and v' are in normal form with respect to R, which implies that $u \approx v$ is false in I and in I_D. □

Lemma 4 *Let $C = \Gamma \rightarrow \Delta, u \approx v$ and D be ground instances of N with $D \succeq^c C$. If $u \approx v$ is true in I_C, then it is also true in I_D and in I.*

Proof. Use the fact that $R_C \subseteq R_D \subseteq R$. □

The above lemmas indicate that the sequence of interpretations I_C, with C ranging over all ground instances of N, preserves the truth of ground clauses:

Corollary 1 *Let C and D be ground instances of N with $D \succeq^c C$. If C is true in I_C, then it is also true in I_D and I.*

The following lemma allows us to restrict our attention to reduced ground instances of clauses in N.

Lemma 5 *Suppose $C\sigma$ is a ground instance of a clause C in N, where $x\sigma$ is reducible by $R_{C\sigma}$, for some variable x occurring in C. Then there is a ground instance $C\tau$ of C, such that (i) $C\sigma \succ^c C\tau$ and (ii) $C\tau$ is true in $I_{C\sigma}$ if and only if $C\sigma$ is true in $I_{C\sigma}$.*

Proof. If $x\sigma \Rightarrow_{R_{C\sigma}} t$, define τ to be the substitution for which $x\tau = t$ and $y\tau = y\sigma$, for all y with $y \neq x$. $\square$

4.2 Redundancy and Saturation

We shall prove that the interpretation I is a model of N, provided N is consistent and saturated, i.e., closed under sufficiently many applications of superposition inference rules.

(Selected) superposition and merging paramodulation are restricted versions of ordinary paramodulation. The ordering constraints and the selection of specific equations in the antecedent of a clause can be thought of as ways of pruning the search space of ordinary paramodulation. While these constraints are local in that they depend on information contained in a given clause, we shall demonstrate that the search space can be further decreased by certain non-local restrictions which are based on the concept of redundancy (of clauses and inferences). Intuitively, a clause is redundant if it does not contribute to the definition of the intended model I of N. An inference is redundant if either one of its premises is redundant or else its conclusion does not contribute to the definition of the intended model I of N. Saturation then means that all non-redundant inferences have been computed.

The following definitions refer to a given set of clauses N and the interpretation I constructed from N. If C is a ground clause, we denote by N_C the set of all ground instances C' of N for which $C \succ^c C'$.

Definition 4 A ground instance C of a clause in N is said to be *redundant* if it is true in I_C. A clause in N is called redundant if all its ground instances are redundant.

By Corollary 1, redundant clauses are true in I. The interpretation I is completely determined by productive clauses, which are non-redundant.

Definition 5 An inference π from ground instances of N is said to be *redundant* if either one of its premises is redundant or else its conclusion is true in I_C, where C is the maximal premise of π. An inference from N is redundant if all its ground instances are redundant. We say that N is *saturated on N'* if every ground instance of an inference from N, the premises of which are in N', is redundant.

For instance, we will deal with sets N that are saturated on N_C, for some ground clause C. Evidently, if N is saturated on N', then it is also saturated on any subset of N'. Also, N is simply called saturated if all ground instances of inferences from N are redundant. The essential properties of saturated sets are given in the following lemma.

Lemma 6 *Let N be a set of clauses saturated on N' (with respect to some inference system $\mathcal{E}_S$ or $\mathcal{P}_S$). Suppose $C = \Gamma \rightarrow s \approx t, \Delta$ is a non-redundant ground instance of some clause D in N, where $N_C \cup \{C\} \subseteq N'$, $s \succ t$, the term s is irreducible by R_C, and $s \approx t$ is a maximal occurrence of an equation in C. Then (i) $C = D\sigma$ is a reduced ground instance of D with respect to R_C; (ii) C contains no selected equation; (iii) C produces $s \approx t$; and (iv) $\Gamma \subseteq I$ and $\Delta \cap I = \emptyset$.*

Proof. The proof is by induction on the clause ordering $\succ^c$. Suppose N is saturated on N'. Let $C = \Gamma \rightarrow s \approx t, \Delta$ be a non-redundant clause, such that $N_C \cup \{C\} \subseteq N'$, $s \succ t$, the term s is irreducible by R_C, and $s \approx t$ is a maximal occurrence of an equation in C. Since C is non-redundant, we have $\Gamma \subseteq I_C$ and $(\Delta \cup \{s \approx t\}) \cap I_C = \emptyset$. We have to prove that C satisfies properties (i)-(iv). Let us assume properties (i)-(iv) hold for all suitable clauses C' with $C \succ^c C'$.

(i) Suppose $C = D\sigma$ is not reduced, i.e., $x\sigma$ is reducible by R_C, for some variable x occurring in D. (Note that $x\sigma$ can not occur in s, as s is irreducible by R_C.) By Lemma 5 there exists a ground instance $C' = D\tau = \Gamma' \rightarrow \Delta', s \approx t'$, such that $C \succ^c C'$, $\Gamma' \subseteq I_C$, and $(\Delta' \cup \{s \approx t'\}) \cap I_C = \emptyset$. By Lemma 2 we have $R_{C'} = R_C$, which implies that C' is false in $I_{C'}$ and hence non-redundant. Since $C \succ^c C'$, we may use the induction hypothesis to infer that properties (i)-(iv) hold for C'. In particular, C' produces some equation $s \approx t''$. Since $E_{C'} \subseteq R_C$, this contradicts the assumption that s is irreducible by R_C.

For the remaining part of the proof, we assume that C is a reduced ground instance of D with respect to R_C.

(ii) Suppose C contains a selected equation. We distinguish two subcases.

If C is a clause $\Gamma', u \approx u \rightarrow \Delta, s \approx t$, where $u \approx u$ is selected, then $C' = \Gamma' \rightarrow \Delta, s \approx t$ may be obtained from C by selective resolution. Since N is saturated on $N_C \cup \{C\}$, the resolution inference has to be redundant. Thus C' has to be true in I_C, which contradicts that $\Gamma' \subseteq I_C$ and $(\Delta \cup \{s \approx t\}) \cap I_C = \emptyset$.

If C is a clause $\Gamma', u \approx v \rightarrow \Delta, s \approx t$, where $u \succ v$ and $u \approx v$ is selected, then $u \Downarrow_{R_C} v$ and therefore u is reducible by R_C. Let $C' = \Lambda \rightarrow \Pi, w \approx w'$, where $C \succ^c C'$, be a non-redundant clause that produces the rule $w \approx w'$, where w is a subterm of u. Using the induction hypothesis, we may infer that the clause contains no selected equations, $\Lambda \subseteq I_C$, $\Pi \cap I_C = \emptyset$, and $w \approx w' \in I_C$. Consider the inference

$$\frac{\Lambda \rightarrow \Pi, w \approx w' \quad \Gamma, u[w] \approx v \rightarrow \Delta, s \approx t}{\Lambda, \Gamma, u[w'] \approx v \rightarrow \Pi, \Delta, s \approx t}$$

by selective paramodulation. Since C is a reduced ground instance with respect to R_C, this inference is a ground instance of an inference from N. Since N is saturated on $N_C \cup \{C\}$, the inference has to be redundant. That is, the conclusion C'' has to be true in I_C, which contradicts that $\Lambda \cup \Gamma \cup \{u[w'] \approx v\} \subseteq I_C$ and $(\Pi \cup \Delta \cup \{s \approx t\}) \cap I_C = \emptyset$.

For the remaining part of the proof, we assume that C contains no selected equations.

(iii) If Δ is of the form $\Delta', s \approx t$, then the clause $\Gamma \rightarrow \Delta', s \approx t$ can be obtained from C by ordered factoring, whereas $\Gamma, t \approx t \rightarrow \Delta', s \approx t$ can be obtained by equality factoring. Both clauses are false in I_C. However, since N is saturated with respect to $\mathcal{E}_S$ or $\mathcal{P}_S$, at least one of the two clauses has to be true in I_C, which is a contradiction. We may therefore assume that $s \approx t$ does not occur in Δ, which implies that C produces $s \approx t$.

(iv) First observe that $\Gamma \subseteq I_C \subseteq I$. Now suppose Δ contains an equation $u \approx v$ which is true in I. Since $\Delta \cap I_C = \emptyset$, we have $u \approx v \in I \setminus I_C$, which is only possible if $s = u$ and $t \Downarrow_{I_C} v$. Since $t \succ v$, the term t is reducible by R_C. We distinguish two cases.

If N is saturated on $N_C \cup \{C\}$ with respect to $\mathcal{E}_S$, then the clause $\Gamma, t \approx v \rightarrow \Delta', s \approx v$, which can be obtained from C by equality factoring, has to be true in I_C. This contradicts that $\Gamma \cup \{t \approx v\} \subseteq I_C$ and $\Delta \cap I_C = \emptyset$.

Suppose N is saturated $N_C \cup \{C\}$ with respect to $\mathcal{P}_S$. Since t is reducible by R_C, there exists a clause $C' = \Lambda \rightarrow \Pi, w \approx w'$, $C \succ^c C'$, that produces a rule $w \approx w'$, where w is a subterm of t. Consider the inference

$$\frac{\Lambda \rightarrow \Pi, w \approx w' \quad \Gamma \rightarrow \Delta', s \approx v, s \approx t[w]}{\Lambda, \Gamma \rightarrow \Pi, \Delta', s \approx v, s \approx t[w']}$$

by merging paramodulation. The conclusion of this inference has to be true in I_C, which again leads to a contradiction.

In sum, we may conclude that $\Delta \cap I = \emptyset$. $\square$

Lemma 7 *Suppose N is saturated on N' and does not contain the empty clause. If C is a ground instance of N and $(N_C \cup \{C\}) \subseteq N'$, then C is true in $(R_C \cup E_C)^*$.*

Proof. Suppose N is saturated on a set N' and does not contain the empty clause. Let C be a ground instance of N, such that $(N_C \cup \{C\}) \subseteq N'$.

If C is redundant or produces an equation, then it is true in $(R_C \cup E_C)^*$. Let us therefore assume that C is neither redundant nor productive. In other words, $E_C = \emptyset$ and C is false in I_C. Using Lemma 5 we may infer that C is a reduced ground instance of N with respect to R_C. Also, C can not be the empty clause. Let s denote the maximal term in C.

(i) Suppose C is a clause $\Gamma', s \approx s \rightarrow \Delta$, where $s \approx s$ is either a maximal or a selected occurrence of an equation. The clause $C' = \Gamma' \rightarrow \Delta$ can be obtained from C by selective or equality resolution. Since N is saturated on $N_C \cup \{C\}$, the inference has to be redundant. Thus C' has to be true in I_C, which contradicts that $\Gamma' \subseteq I_C$ and $\Delta \cap I_C = \emptyset$.

(ii) Suppose C is a clause $\Gamma', s \approx t \rightarrow \Delta$, where $s \succ t$ and $s \approx t$ is either a maximal or a selected occurrence of an equation. Then $s \approx t \in I_C$ and s is reducible by R_C. Let $D = \Lambda \rightarrow u \approx v, \Pi$ be a clause that produces the rule $u \approx v$, where $u \succ v$ and u is a subterm of s. Then $C \succ^c D$ and using Lemma 6 we may infer that $\Lambda \subseteq I_C$, $u \approx v \in I_C$, $\Pi \cap I_C = \emptyset$, and D contains no selected equations.

Consider the inference

$$\frac{\Lambda \rightarrow u \approx v, \Pi \quad \Gamma', s[u] \approx t \rightarrow \Delta}{\Gamma', \Lambda, s[v] \approx t \rightarrow \Delta, \Pi}$$

by (selective) superposition. Since N is saturated on $N_C \cup \{C\}$, the conclusion C' of this inference has to be true in I_C. This contradicts that $(\Gamma' \cup \Lambda) \subseteq I_C$, $s[v] \approx t \in I_C$ and $(\Delta \cup \Pi) \cap I_C = \emptyset$.

(iii) Suppose C is $\Gamma \rightarrow \Delta', s \approx t$, where $s \succ t$, $s \approx t$ is a maximal occurrence of an equation in C, and the term s is reducible by R_C. If $s \approx t$ occurs in Δ' we can derive a contradiction again using the fact that N is in particular saturated (on $N_C \cup \{C\}$) by ordered factoring. Hence $s \approx t$ is a strictly maximal occurrence in C. Let $D = \Lambda \rightarrow u \approx v, \Pi$, where $C \succ^c D$, be a clause that produces the rule $u \approx v$, where $u \succ v$ and u is a subterm of s. Using Lemma 6 we may infer that $\Lambda \subseteq I_C$, $u \approx v \in I_C$, $\Pi \cap I_C = \emptyset$, and D contains no selected equations.

Consider the inference

$$\frac{\Lambda \to u \approx v, \Pi \quad \Gamma \to \Delta', s[u] \approx t}{\Gamma, \Lambda \to \Delta', \Pi, s[v] \approx t}$$

by right superposition. Saturation of N on $N_C \cup \{C\}$ implies that the conclusion C' of this inference has to be true in I_C. This contradicts that $(\Gamma \cup \Lambda) \subseteq I_C$, $s[v] \approx t \notin I_C$, and $(\Delta' \cup \Pi) \cap I_C = \emptyset$. □

Corollary 2 *If N is saturated, then every non-redundant ground instance of N is productive.*

Proof. Suppose C is a non-redundant ground instance of N, i.e., C is false in I_C. By Lemma 7 C is true in $(R_C \cup E_C)^*$, which implies that $E_C \neq \emptyset$. In other words, C produces an equation. □

Theorem 1 *A saturated set of clauses N is consistent if and only if it does not contain the empty clause.*

Proof. If N contains the empty clause, then it has no model. On the other hand, if N is saturated and consistent, then by Lemma 7 every ground instance C of N is true in $(R_C \cup E_C)^*$ and hence true in I. In other words, I is a model of N. □

Let us remark that while we have assumed that the reduction ordering $\succ$ is complete (i.e., total on ground terms), it is sufficient to require that $\succ$ be completable (i.e., contained in some complete ordering $>$). For if a set N of clauses is saturated with respect to some inference system $\mathcal{E}_S^{\succ}$ (or $\mathcal{P}_S^{\succ}$), then it is also saturated with respect to the more restrictive system $\mathcal{E}_S^{>}$ (or $\mathcal{P}_S^{>}$). Therefore our results apply not only to complete orderings, such as certain lexicographic path orderings, but to all completable ordering, for instance, all recursive path orderings.

We conclude this section with a remark on subsumption.

Definition 6 A clause $C = \Gamma \to \Delta$ is said to *subsume* a clause $D = \Lambda \to \Pi$ if there exists a substitution σ such that $\Gamma\sigma \subseteq \Lambda$ and $\Delta\sigma \subseteq \Pi$. We say that C *properly subsumes* D if C subsumes D but not vice versa.

Proposition 1 *If $N \cup \{C\}$ is a saturated and consistent set of clauses, where C subsumes D, then $N \cup \{C, D\}$ is also saturated.*

Proof. Let I be the interpretation constructed from $N \cup \{C, D\}$. Also, let N' be the set of all ground instances of $N \cup \{C, D\}$ and N'' be the set of all ground instances of $N \cup \{C\}$. It suffices to show that all clauses in $N' \setminus N''$ are redundant. Let $D\sigma$ be a clause in $N' \setminus N''$. Since C subsumes D, the clause D can be written as $\Gamma, \Lambda \to \Delta, \Pi$, where $C\tau = \Gamma \to \Delta$, for some τ. Since $D\sigma \notin N''$, we have $D\sigma \succ^c C\tau\sigma$. Since $N \cup \{C\}$ is saturated, $C\tau\sigma$ is true in $I_{D\sigma}$, which implies that $D\sigma$ is also true in $I_{D\sigma}$. Hence $D\sigma$ is redundant. □

4.3 Modular Redundancy Criteria

Clausal theorem proving can be interpreted as a process of constructing saturated sets of clauses. However, the notion of redundancy we have introduced above is too general to be directly applicable to theorem proving. In this section we describe sufficient conditions for redundancy that cover all simplification and deletion techniques common in rewrite-based theorem proving.

Definition 7 Let N be a set of clauses and C be a ground clause (not necessarily a ground instance of N). We call C *composite* with respect to N, if there exist ground instances $C_1, \ldots, C_k$ of N such that $C_1, \ldots, C_k \models C$ and $C \succ^c C_j$, for all j with $1 \leq j \leq k$. A non-ground clause is called composite if all its ground instances are composite.[7]

Lemma 8 *If a clause C is composite with respect to N, then for every ground instance $C\sigma$ there exist non-composite ground instances $C_1, \ldots, C_k$ of N such that $C_1, \ldots, C_k \models C\sigma$ and $C\sigma \succ^c C_j$, for all j with $1 \leq j \leq k$.*

Proof. Let C be a clause that is composite with respect to N and let $C\sigma$ be a ground instance of C. Furthermore, let $N' = \{C_1, \ldots, C_k\}$ be a minimal set of ground instances of N with respect to $\succ^c_{mul}$, such that $C_1, \ldots, C_k \models C\sigma$ and $C\sigma \succ^c C_j$, for all j. We claim that all clauses C_j are non-composite. For if some clause C_j is composite with respect to N, then there exists a set $N'' = \{D_1, \ldots, D_n\}$ of ground instances of clauses in N, such that $D_1, \ldots, D_n \models C_j$. But then $(N' \setminus \{C_j\}) \cup N'' \models C\sigma$ and $N' \succ^c_{mul} (N' \setminus \{C_j\}) \cup N''$, which contradicts the minimality assumption about N'. □

Lemma 9 *Let C be a composite ground instance of some clause in N. If N is saturated on N_C and does not contain the empty clause, then C is redundant.*

Proof. Let $C_1, \ldots, C_k$ be ground instances of N, such that $C_1, \ldots, C_k \models C$ and $C \succ^c C_j$, for all $1 \leq j \leq k$. We may use Lemma 7 to infer that each clause C_j is true in $(R_{C_j} \cup E_{C_j})^*$ and hence true in I_C. Thus C is true in I_C and hence redundant. □

Definition 8 A ground inference π with conclusion B is called *composite* with respect to N if either some premise is composite, or else there exist ground instances $C_1, \ldots, C_k$ of N such that $C_1, \ldots, C_k \models B$ and $C \succ^c C_j$, for all j with $1 \leq j \leq k$, where C is the maximal premise of π. A non-ground inference is called composite if all its ground instances are composite.

Lemma 10 *Let π be a composite ground instance of an inference from N with maximal premise C. If N is saturated on N_C and does not contain the empty clause, then π is redundant.*

Proof. Suppose π is a composite ground instance of an inference from N with maximal premise C and conclusion B, where N is saturated on N_C. We may use Lemma 9 to infer that π is redundant whenever some premise is composite. If all premises are non-redundant, then there exist ground instances $C_1, \ldots, C_k$ of N such that $C_1, \ldots, C_k \models B$ and $C \succ^c C_j$, for all j with $1 \leq j \leq k$. Since N is saturated on N_C and $C \succ^c C_j$, for all j, each clause C_j is true in I_C. Hence B is true in I_C, which shows that the inference π is redundant. □

[7] Again, a more refined way of comparing substitution instances of clauses in N might be based on pairs of clauses and substitutions and involve the subsumption ordering on clauses.

Lemma 11 *(i) If $N \subseteq N'$, then any inference or clause which is composite with respect to N is also composite with respect to N'.*

(ii) If $N \subseteq N'$ and all clauses in $N' \setminus N$ are composite with respect to N', then any inference or clause which is composite with respect to N' is also composite with respect to N.

Proof. Part (i) is obvious; for part (ii) use Lemma 8. □

The lemma shows that compositeness with respect to a set N is preserved if clauses are added to N or if *composite* clauses are deleted. The concept of compositeness thus provides a useful basis for simplification and deletion in a theorem prover.

5 Theorem Proving with Simplification and Deletion

We next consider the problem of constructing a saturated set from a given set of clauses N.

5.1 Theorem Proving Derivations

A theorem prover computes derivations using the following two inference rules on sets of clauses:

$$\textbf{Deduction}: \quad \frac{N}{N \cup \{C\}} \quad \text{if } N \models C$$

$$\textbf{Deletion}: \quad \frac{N \cup \{C\}}{N} \quad \text{if } C \text{ is composite with respect to } N \cup \{C\}$$

Deduction adds clauses that logically follow from given clauses; deletion eliminates composite clauses.

Deduction of a clause D to N which triggers a subsequent deletion of another clause C represents a simplification. If D is needed to prove the compositeness of C, it will be smaller than C with respect to $\succ^c$, so that we have a derived inference

$$\textbf{Simplification}: \quad \cfrac{\cfrac{N \cup \{C\}}{N \cup \{C, D\}}}{N \cup \{D\}}$$

Note that simplification may require the deduction of (logically sound) clauses other than those that can be obtained by $\mathcal{E}_S$ or $\mathcal{P}_S$. This is also the reason why we have not restricted the above deduction to a superposition calculus, but allow for the application of any sound inference rule.

For example, let $\rightarrow s \approx t$ and $\rightarrow u \approx v[s]$ be two unit clause, where $s \succ t$ and $u \succ v$. If we deduce $\rightarrow u \approx v[t]$ (by paramodulation, not superposition), then the clause $\rightarrow u \approx v[s]$ becomes composite and hence can be deleted.

Definition 9 A (finite or countably infinite) sequence $N_0, N_1, N_2, \ldots$ of sets of clauses is called a *theorem proving derivation* if each set N_{i+1} can be obtained from N_i by deduction or deletion. The set $N_\infty = \bigcup_j \bigcap_{k \geq j} N_k$ is called the *limit* of the derivation. Clauses in N_∞ are called *persisting.*

Definition 10 A theorem proving derivation is called *fair* (with respect to $\mathcal{E}_S$ or $\mathcal{P}_S$) if every inference from N_∞ is composite with respect to $\bigcup_j N_j$.

A fair derivation can be constructed, for instance, by systematically adding conclusions of non-composite inferences from persisting clauses. It is important to notice that because of the monotonicity of the ground inferences with respect to $\succ^c$ — the conclusions are smaller than the maximal premises — adding the conclusion of an inference makes the inference to become composite afterwards.

Definition 11 A set of clauses N is called *complete* (with respect to $\mathcal{E}_S$ or $\mathcal{P}_S$) if all inferences from N are composite with respect to N.

For instance, any set containing the empty clause is complete. Another example of a complete set of clauses is the theory of a total order p:

$$\begin{array}{rcl} & \rightarrow & p(x,x) \\ & \rightarrow & p(x,y), p(y,x) \\ p(x,y), p(y,z) & \rightarrow & p(x,z) \\ p(x,y), p(y,x) & \rightarrow & x \approx y \end{array}$$

The proof of completeness is rather tedious and proceeds by case analysis on the inequalities with respect to $\succ$ between the terms to be substituted for variables. (Any total reduction ordering on ground terms is suitable. We assume that $p(s,t) \succ p(u,v)$ if and only if either $s \succ u$ or else $s = u$ and $t \succ v$.) For example, consider the ordered resolution inference

$$\frac{p(x,y), p(y,z) \rightarrow p(x,z) \quad p(x,z), p(z,x) \rightarrow x \approx z}{p(x,y), p(y,z), p(z,x) \rightarrow x \approx z}$$

If any two of the variables x, y, and z are instantiated by the same ground term, the inference is composite as one of the premises would be composite. If pairwise distinct ground terms s, t, and u are substituted for x, y, and z, respectively, then the inference is ordered only if $s \succ u \succ t$. But then

$$\begin{array}{rcl} p(t,u), p(u,s) & \rightarrow & p(t,s), \\ p(s,t), p(t,s) & \rightarrow & s \approx t, \\ p(u,s), p(s,t) & \rightarrow & p(u,t), \\ p(t,u), p(u,t) & \rightarrow & t \approx u \end{array} \models p(s,t), p(t,u), p(u,s) \rightarrow s \approx u \, .$$

In other words, the conclusion of the ground inference logically follows from ground instances of clauses simpler than the maximal premise. Thus the inference is composite.

Note that this case analysis on the ordering between variables—a technique that has also been described by Martin and Nipkow (1989) in the context of ordered completion of equations—is independent of the signature, that is, of any additional function or predicate symbols that might exist besides p.

Fairness, completeness, and saturation are related in the following way.

Lemma 12 *If $N_0, N_1, N_2, \ldots$ is a fair theorem proving derivation (with respect to $\mathcal{E}_S$ or $\mathcal{P}_S$), then N_∞ is complete and every clause C in $(\bigcup_j N_j) \setminus N_\infty$ is composite with respect to N_∞.*

Proof. If C is a clause in $(\bigcup_j N_j) \setminus N_\infty$ then it is composite with respect to some set N_j and hence composite with respect to $\bigcup_j N_j$. We may use Lemma 11 to infer that (i) all clauses in in $(\bigcup_j N_j) \setminus N_\infty$ are composite with respect to N_∞, and (ii) every ground instance of an inference from N_∞ is composite with respect to N_∞. □

Lemma 13 *Any complete set of clauses that does not contain the empty clause is saturated.*

Proof. Suppose N is complete set of clauses and does not contain the empty clause. We shall prove that N is saturated on $N_C \cup \{C\}$, for all ground instances C of N.

Let C be minimal ground instance of N with respect to the clause ordering $\succ^c$, such that N is not saturated on $N_C \cup \{C\}$. Then N is saturated on N_C and there exists some non-redundant ground instance π of an inference from N, the maximal premise of which is C. Since N is complete, the inference π has to be composite. Using Lemma 10 we may infer that π is redundant, which is a contradiction. □

Theorem 2 *Let $N_0, N_1, N_2, \dots$ be a fair theorem proving derivation (with respect to $\mathcal{E}_S$ or $\mathcal{P}_S$). If $\bigcup_j N_j$ does not contain the empty clause, then N_∞ is saturated and N_0 is consistent.*

Proof. By fairness, the set N_∞ is complete. If it does not contain the empty clause, then by Lemma 13 it is saturated. Using Lemma 12 we may infer that the interpretation I constructed from N_∞ is a model of $\bigcup_j N_j$. □

5.2 Simplification and Deletion Techniques

Most simplification techniques proposed for theorem proving are specific tests for compositeness. We discuss some of these.

If C is a clause and N is a set of clauses, let in the following N_C denote the set of all (ground or non-ground) instances $D\sigma$ of clauses D in N such that $C \succ^c D\sigma$. Let us write $[N] \models C$ if $N\sigma \models C\sigma$, for all ground instances $C\sigma$ and $N\sigma$ of C and N, respectively. Note that $[N] \models C$ implies that C is composite in N, if $C \succ^c D$, for all clauses D in N. ($N \models C$ and $C \succ^c D$, for all clauses D in N, does not necessarily imply that C is composite in N. It may be the case that although $C \succ^c D$ for each clause D in N, to prove that some ground instance $C\sigma$ follows from N requires to use a ground instance $D\tau$ of N with $D\tau \succ^c C\sigma$.)

Elimination of redundant atoms. Let $C = \Gamma, u \approx v \to \Delta$ be a clause in N. If $N \models \Gamma \to u \approx v, \Delta$, then $N \models \Gamma \to \Delta$, so that C can be simplified to $\Gamma \to \Delta$. A particular case is the elimination of multiple occurrences of atoms in the antecedent. For example, if $C = \Gamma, u \approx v, u \approx v \to \Delta$, then the clause $\Gamma, u \approx v \to u \approx v, \Delta$ is a tautology and hence trivially implied by N. Redundant atoms in the succedent can be eliminated in a similar way.

Case analysis. The first step in a case analysis consists of splitting a clause $C = \Gamma \to \Delta$ into n clauses $C_i = A_i, \Gamma \to \Delta$, where $[N_C] \models \Gamma \to A_1, \dots, A_n, \Delta$. Each clause C_i logically follows from C and hence can be deduced. If in addition there exist clauses D_i, such that $N_C \models D_i \supset C_i$ and $C \succ^c D_i$, for all i with $1 \le i \le n$, then $[N_C \cup \{D_1, \dots, D_n\}] \models C$, which indicates that C is composite in $N \cup \{D_1, \dots, D_n\}$.

In practice, simplification is usually employed to construct clauses D_i that are logically equivalent to the respective clauses C_i, but simpler than C. The possibility of such simplification depends to some extent on the choice of the "cases" C_i. The case analysis can also be applied recursively to the clauses C_i. Contextual rewriting (Zhang and Rémy 1985, Navarro 1987) or splitting of clauses (Ganzinger 1987b) are particular instances of a case analysis.

Contextual reductive rewriting. Let $C = \Gamma \rightarrow \Delta, s \approx t$ and $D = \Lambda, A[u] \rightarrow \Pi$ (or $D = \Lambda \rightarrow \Pi, A[u]$) be clauses in N and σ be a substitution, such that (i) u is $s\sigma$, (ii) $s\sigma \succ t\sigma$, (iii) $D \succ^c C\sigma$, (iv) $[N_D] \models \Lambda \rightarrow B$, for all equations $B \in \Gamma\sigma$, and (v) $[N_D] \models B \rightarrow \Pi$, for all equations $B \in \Delta\sigma$. Then $D' = \Lambda, A[t\sigma] \rightarrow \Pi$ (or $D' = \Lambda \rightarrow \Pi, A[t\sigma]$) logically follows from N and moreover $[N_D \cup \{D'\}] \models D$. Since $D \succ^c D'$ this indicates that D is composite (and hence can be deleted) after D' has been deduced.

Replacing D by D' generalizes simplification by contextual reductive conditional rewriting as described by Ganzinger (1987a) for a completion procedure for conditional equations, and of course also covers ordinary (unconditional) rewriting. (In this context by a reductive conditional rewrite rule one means a clause $C = \Gamma \rightarrow \Delta, s \approx t$, where s is a strictly maximal term. If the term s matches a proper subterm of a clause D, then $C\sigma \in N_D$.) The contextual aspect of the simplification is expressed in conditions (iv) and (v) where an equation B needs to be true only for those substitutions that make Λ true and Π false. In practice, proofs of $[N_D] \models \Lambda \rightarrow u \approx v$ may be conducted by reductive conditional rewriting with (instances of) clauses in N_C, using equations in the (Skolemized) antecedent Λ as additional rewrite rules.

6 Refutation of Goals

In this section we consider situations in which a fair theorem proving derivation from some finite initial set of clauses N_0 terminates after finitely many steps without encountering an inconsistency, so that for some k, $N_\infty = N_k$ is a finite complete set of clauses. Because of the powerful concept of redundancy which we have introduced before there is reason to believe that termination of completion is not unusual in practice.

Finite, complete and consistent sets N of clauses will also be called *programs*. A formula

$$\neg G \;=\; \exists \vec{x}\ (A_1 \wedge \ldots \wedge A_n \wedge \neg B_1 \wedge \ldots \wedge \neg B_k)$$

is a logical consequence of a program N if and only if $N \cup \{G\}$ is inconsistent, where G is the clause $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_k$ (also called a *goal*).

The search for a refutation of $N \cup \{G\}$ may be simpler and more efficient for several reasons than the search for a refutation in general.

- Since N is complete, inferences between clauses in N are composite and remain composite at any step of a theorem proving derivation from $N \cup \{G\}$. In other words, if N is complete and consistent, $\{G\}$ forms a *set of support* for some refutation of $N \cup \{G\}$, if $N \cup \{G\}$ is inconsistent. Notice that this is not true in general for arbitrary sets N of clauses.[8] Also,

[8] In general, the set-of-support restriction is not refutationally complete with paramodulation, or with ordered resolution. Snyder and Lynch (1991) describe a "lazy paramodulation" calculus that is complete with set of support.

clauses in N with selected equations need often not be considered during the refutation of goals.

- Under certain additional assumptions it may be sufficient to compute only so-called N*-linear* superposition inferences

$$\frac{C \quad D}{B}$$

where $C \in N$ and $D \notin N$. For example, linear SLD-resolution and linear conditional narrowing are refutationally complete for certain Horn clause programs.

- Theorem provers usually employ some backtracking mechanism. If an inference system is *don't care nondeterministic*, in the sense that a refutation can be constructed regardless of the order in which inference rules are applied, then backtracking is not needed. For example, rewriting with convergent rewrite systems as it is employed in many completion procedures is don't care nondeterministic.

  In certain cases it may even be decidable whether a goal G is refutable, e.g., if the inferences in any refutation of a goal are strictly decreasing in that the conclusion is smaller than some premise with respect to a given well-founded ordering. Thus the validity problem in an equational theory represented by a finite convergent rewrite system is decidable.

The completeness of linear superposition for refutation of goals can be proved for programs with certain syntactic properties.

Definition 12 A *quasi-Horn clause* is a clause $\Gamma, \Sigma \rightarrow \Delta$ or $\Gamma, \Sigma \rightarrow \Delta, s \approx t$, where Γ contains only function equations, $\Sigma \cup \Delta$ contains only predicate equations, and $s\sigma \approx t\sigma$ is strictly maximal in $\Sigma\sigma \cup \Delta\sigma$, for all ground substitutions σ.

Quasi-Horn clause programs correspond to what are sometimes called Horn clause specifications over "built-in Booleans." In such programs predicates are defined by clauses $\Gamma \rightarrow \Delta$ with no function equations in the succedent, whereas functions are defined by clauses $\Gamma \rightarrow \Delta, s \approx t$. The fact that predicates in $\Gamma \rightarrow \Delta, s \approx t$ have to be simpler (with respect to the ordering $\succ$) than the function equation $s \approx t$, generalizes the idea of a hierarchical specification over built-in Booleans.

The following example defines a function for ordered insertion where the inequalities are expressed by a predicate $<$ and a derived predicate $\geq$. Any lexicographic path ordering with a precedence $\succ$ in which *insert* precedes $>$ and $\geq$ will ensure the required syntactic properties.

$$
\begin{aligned}
x' < 0 &\rightarrow && (1)\\
&\rightarrow \quad 0 < x' && (2)\\
x < y &\rightarrow \quad x' < y' && (3)\\
x' < y' &\rightarrow \quad x < y && (4)\\
x < y,\ x \geq y &\rightarrow && (5)\\
&\rightarrow \quad x < y,\ x \geq y && (6)\\
&\rightarrow \quad insert(nil, x) = cons(x, nil) && (7)\\
&\rightarrow \quad x \geq y,\ insert(cons(y, l), x) = cons(x, cons(y, l)) && (8)\\
x \geq y &\rightarrow \quad insert(cons(y, l), x) = cons(y, insert(l, x)) && (9)
\end{aligned}
$$

It is evident from the syntactic restrictions that merging paramodulation cannot be applied to quasi-Horn clauses and that equality factoring is essentially identical to ordered factoring. Quasi-Horn clauses with a funtion equation in the succedent cannot be premises of ordered resolution and ordered factoring inferences. Moreover, if D has no function equation in its succedent and B is obtained from C and D by one application of (selective) superposition, then B has no function equation in the succedent either.

Lemma 14 *Let $N \cup M$ be a complete set of quasi-Horn clauses with respect to an inference system $\mathcal{E}_S$, where each clause in M contains selected equations, and let $G_1, \ldots, G_n$ be clauses with no function equation in the succedent. If $N \cup M \cup \{G_1, \ldots, G_n\} = N_0, N_1, \ldots$ is a theorem proving derivation in which no deduction step adds a clause with a function equation in the succedent, then any (selective) superposition inference from N_∞ is either N-linear or composite in N_0.*

Proof. Let $N_0, N_1, \ldots$ be a derivation where no clause with a function equation in the succedent is ever deduced. Thus the only clauses in $\bigcup_j N_j$ with function equations in the succedent are those in $N \cup M$. Since clauses in M contain selected equations, the first premise of any (selective) superposition inference from N_∞ has to be in N. If the second premise is in $N \cup M$, then the inference is composite, by the completeness of $N \cup M$. If the second premise is not in $N \cup M$, the inference is N-linear. □

The lemma indicates that for quasi-Horn programs the refutation of goals without function equations in the succedent is linear with respect to the equality part of the logic. Ordered resolution, which covers the non-functional aspects of the program, is still nonlinear (but fortunately is a rather restricted form of resolution). The clauses in M, which contain selected equations, might be called *nonoperational*. They have presumedly been used in the construction of the complete set $N \cup M$, but are not needed for the refutation of goals.

Corollary 3 *Let $N \cup M$ be a complete set of Horn clauses with respect to some inference system $\mathcal{E}_S$, where each clause in M contains selected equations. Moreover, let G be a clause with empty succedent and let $N \cup M \cup \{G\} = N_0, N_1, \ldots$ be a theorem proving derivation in which no deduction step adds a clause with a non-empty succedent. Then N_∞ is complete with respect to $\mathcal{E}_S$ if and only if it is complete with respect to N-linear (selective) superposition, N-linear ordered resolution, and selective and equality resolution.*

Proof. By the completeness of $N \cup M$, any inference is composite if both premises are in $N \cup M$. Since any two-premise inference rule in $\mathcal{E}_S$ requires at least one premise with a non-empty succedent, the only non-composite inferences with two premises have to be N-linear. Selective and equality resolution require only one premise. □

The corollary indicates that certain ordered variants of conditional narrowing are complete for refuting the negations of a conjunction of equations in a complete set of conditional equations. Note that there are no restrictions about variables. The slightly weaker result obtained by Bertling and Ganzinger (1989) did not admit conditional rewrite rules with variables as their left side.

Let us next consider refutation of ground goals.

Definition 13 A clause $C = \Gamma \to \Delta$ is called *universally reductive* if either the succedent Δ is empty, or else Δ can be written as Δ', A such that (i) all variables of C also occur in A, (ii) $C\sigma$ is reductive for $A\sigma$, for all ground substitutions σ, and (iii) if A is a function equation $s \approx t$, then all variables of A occur in s and $s\sigma \succ t\sigma$, for all ground substitutions σ.

Theorem 3 *Let $N \cup M$ be a finite complete set of quasi-Horn clauses with respect to an inference system $\mathcal{E}_S$, where each clause in M contains selected equations, and let $G_1, \ldots, G_n$ be ground clauses with no function equation in the succedent. If all clauses in N are universally reductive, then $N \cup M \models \neg(G_1 \wedge \ldots \wedge G_n)$ is decidable by N-linear (selective) superposition, ordered factoring, selective and equality resolution, and (non-linear) ordered resolution.*

Proof. The linearity property follows from Lemma 14. The stated requirements ensure that any non-composite inference is ground, so that the conclusion is smaller with respect to the well-founded ordering $\succ^c$ than the maximal premise. As a consequence a finite fair derivation can be obtained by applying the given inference rules. $\square$

In the example above, let M be the set consisting of clause (4) (i.e., assume that the condition of clause (4) is selected) and let N be the set of all remaining clauses. Then N and M satisfy the requirements of Theorem 3, hence the ground theory as specified by the example is decidable.

The above results do not cover *goal solving*, i.e., the process of finding substitutions that refute the goal. In the case of Horn clauses, all irreducible substitutions solving a goal can be enumerated by ordered conditional narrowing. This does not hold for quasi-Horn clauses, in general, as shown by the following example:

$$
\begin{array}{rcl}
 & \to & p,\ q \\
p & \to & a \approx b \\
q & \to & c \approx d \\
 & \to & f(x,x,y,z) \approx z \\
 & \to & f(x,y,z,z) \approx z
\end{array}
$$

where $a \succ b \succ c \succ d \succ p \succ q$ and p and q are predicate constants. This set of quasi-Horn clauses is complete, but the solution $\{x \mapsto h(a), y \mapsto h(b), u \mapsto h(c), v \mapsto h(d)\}$ for $f(x,y,u,v) \approx v$ cannot be obtained from the given axioms and the goal

$$f(x,y,u,v) \approx v \to$$

by paramodulation if the functional-reflexive axioms are not available. The difficulty arises from disjunctions of equations which, as in the example above, can easily be specified via propositional variables.

We conclude this section with a few remarks about "don't care nondeterminism," a frequent case of which occurs when a superposition inference is actually a simplification by contextual rewriting

$$\frac{\Gamma \to s \approx t, \Delta \quad \Lambda, A[s\sigma] \to \Pi}{\Gamma\sigma, \Lambda, A[t\sigma] \to \Delta\sigma, \Pi}$$

such that "$\Gamma\sigma$ is true and $\Delta\sigma$ is false in context $\Lambda \rightarrow \Pi$" (as described formally in Section 5). Here $\Lambda, A[s\sigma] \rightarrow \Pi$ typically is a goal with no function equation in the succedent. Once the new goal $\Gamma\sigma, \Lambda, A[t\sigma] \rightarrow \Delta\sigma, \Pi$ has been deduced, the old goal $\Lambda, A[s\sigma] \rightarrow \Pi$ becomes composite and can be deleted. In other words, no further inference with the old goal are required. If N is a Horn clause program this implies the ground confluence of conditional reductive rewriting.

Theorem 4 *Let $N \cup M$ be a complete set of Horn clauses with respect to an inference system $\mathcal{E}_S$, where each clause in M contains selected equations. Let N_R be a set of universally reductive instances of clauses in N, such that N_R has the same reductive ground instances as N. Then the initial algebras of N_R, N, and $N \cup M$ coincide, and recursive conditional rewriting with N_R is ground convergent.*

Proof. The construction of an interpretation I for a saturated set of clauses K yields the initial model of K, if K is a set of Horn clauses (Bachmair and Ganzinger 1991). By Theorem 6, ground instances of clauses in M are non-productive and therefore do not contribute to the definition of the initial model I of $N \cup M$. Ground instances of N_R are either redundant or productive. Since productive ground instances are reductive and N_R has the same reductive ground instances as N, we may conclude that each productive clause is an instance of a clause in N_R. The convergence of recursive rewriting with N_R is therefore an immediate consequence of the convergence of the set of rules R defining I. $\square$

Note that if $N \cup M$ is also complete with respect to an extension of the given signature (and term ordering) by infinitely many new constants, the last result actually implies convergence on general terms. However, the completeness of $N \cup M$ need not be preserved under such an extension as the compositeness of an inference or clause may depend on the signature.

The clauses in M are inductive theorems of the clauses in N. The theorem thus opens up new ways of completion-based inductive theorem proving for Horn clauses, as it also avoids the problems with the undecidability of inductive reducibility for Horn clauses. This is another interesting application of explicit selection strategies for negative literals. With an appropriate coding, the selection of literals corresponds to the selection of induction variables in more traditional induction techniques. A more elaborate treatment of these ideas is beyond the scope of the present paper.

References

[1] L. Bachmair, 1991. *Canonical equational proofs.* Birkhäuser, Boston.

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

[3] L. Bachmair and H. Ganzinger, 1990. On restrictions of ordered paramodulation with simplification. In *Proc. 10th Int. Conf. on Automated Deduction*, Lect. Notes in Comput. Sci., vol. 449, pp. 427–441, Berlin, Springer-Verlag.

[4] L. Bachmair and H. Ganzinger, 1991. Perfect Model Semantics for Logic Programs with Equality. In *Proc. 8th Int. Conf. on Logic Programming*, pp. 645–659, MIT Press.

[5] H. Bertling and H. Ganzinger, 1989. Completion-time optimization of rewrite-time goal solving. In *Proc. 3rd Int Conf. Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., vol. 355, Berlin, Springer-Verlag.

[6] N. Dershowitz, 1991. A Maximal-Literal Unit Strategy for Horn Clauses. To appear in *Proc. Second Int. Workshop on Conditional and Typed Rewriting Systems*, Lect. Notes in Comput. Sci., Berlin, Springer-Verlag.

[7] H. Ganzinger, 1987. A completion procedure for conditional equations. In S. Kaplan and J.-P. Jouannaud, editors, *Conditional Term Rewriting Systems*, Lect. Notes in Comput. Sci., vol. 308, pp. 62–83, Berlin, Springer-Verlag. Revised version in *J. Symbolic Computation 11*, pp. 51–81.

[8] H. Ganzinger, 1987. Ground term confluence in parametric conditional equational sepcifications. In *STACS'87*, Lect. Notes in Comput. Sci., vol. 247, pp. 286–298, Berlin, Springer-Verlag.

[9] J. Hsiang and M. Rusinowitch, 1987. On word problems in equational theories. In T. Ottmann, editor, *Proc. 14th ICALP*, Lect. Notes in Comput. Sci., vol. 267, pp. 54–71, Berlin, Springer-Verlag.

[10] J. Hsiang and M. Rusinowitch, 1989. Proving refutational completeness of theorem proving strategies: The transfinite semantic Tree method. Submitted for publication.

[11] G. Huet, 1980. Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the ACM*, Vol. 27, pp. 797–821.

[12] D. Knuth and P. Bendix, 1970. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pp. 263–297. Pergamon Press, Oxford.

[13] E. Kounalis and M. Rusinowitch, 1988. On word problems in Horn theories. In E. Lusk and R. Overbeek, editors, *Proc. 9th Int. Conf. on Automated Deduction*, Lect. Notes in Comput. Sci., vol. 310, pp. 527–537, Berlin, Springer-Verlag.

[14] U. Martin and T. Nipkow, 1989. Ordered Rewriting and Confluence. Technical Report, Univ. of Cambridge, Cambridge, U.K.

[15] M. Navarro, 1987. Técnicas de reescritura para especificaciones condicionales.. Dissertation, Universitat Politècnica de Catalunya.

[16] R. Nieuwenhuis and F. Orejas, 1991. Clausal Rewriting. To appear in *Proc. Second Int. Workshop on Conditional and Typed Rewriting Systems*, Lect. Notes in Comput. Sci., Berlin, Springer-Verlag.

[17] M. Rusinowitch, 1991. Theorem proving with resolution and superposition: An extension of the Knuth and Bendix procedure as a complete set of inference rules. *J. Symbolic Computation 11*.

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

[19] W. Snyder and C. Lynch, 1991. Goal directed strategies for paramodulation. In R. Book, editor, *Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., vol. 488, pp. 150–161, Berlin, Springer-Verlag.

[20] L. T. Wos, G. A. Robinson, D. F. Carson, and L. Shalla, 1967. The concept of demodulation in theorem proving. *J. of the ACM*, Vol. 14, pp. 698–709.

[21] H.T. Zhang and J-L. Rémy, 1985. Contextual Rewriting. In J.-P. Jouannaud, editor, *Rewriting Techniques and Applications*, Lect. Notes in Comput. Sci., vol. 202, pp. 46–62, Berlin, Springer-Verlag.