Strict Basic Superposition
and
Chaining

Leo Bachmair
Harald Ganzinger

MPI–I–97–2–011 December 1997

Authors' Addresses

Leo Bachmair
Department of Computer Science
SUNY at Stony Brook, Stony Brook, NY 11794, U.S.A.
email: `leo@cs.sunysb.edu`, web: `www.cs.sunysb.edu/~leo`

Harald Ganzinger
Max-Planck-Institut für Informatik
Im Stadtwald, D-66123 Saarbrücken, Germany
email: `hg@mpi-sb.mpg.de`, web: `www.mpi-sb.mpg.de/~hg`

Publication Notes

A short version of this report has been submitted for publication elsewhere.

Acknowledgements

This research was supported in part by the National Science Foundation under grants INT-9314412 and CCR-9510072, by the German Science Foundation (Deutsche Forschungsgemeinschaft) under grants Ga 261/7-1 and Ga 261/8-1, and by the ESPRIT Basic Research Working Group 22457 (Construction of Computational Logics II).

Abstract

The most efficient techniques that have been developed to date for equality handling in first-order theorem proving are based on superposition calculi. Superposition is a refinement of paramodulation in that various ordering constraints are imposed on inferences. For practical purposes, a key aspect of superposition is its compatibility with powerful simplification techniques. In this paper we solve a long-standing open problem by showing that strict superposition—that is, superposition without equality factoring—is refutationally complete. The difficulty of the problem arises from the fact that the strict calculus, in contrast to the standard calculus with equality factoring, is not compatible with arbitrary removal of tautologies, so that the usual techniques for proving the (refutational) completeness of paramodulation calculi are not directly applicable. We deal with the problem by introducing a suitable notion of *direct rewrite proof* and modifying proof techniques based on candidate models and counterexamples in that we define these concepts, not in terms of semantic truth, but in terms of direct provability. We also introduce a corresponding concept of redundancy with which strict superposition is compatible and that covers most simplification techniques, though not, of course, removal of *all* tautologies. Reasoning about the strict calculus, it turns out, requires techniques known from the more advanced *basic* variant of superposition. Superposition calculi, in general, are parametrized by (well-founded) literal orderings. We prove refutational completeness of strict basic superposition for a large class of such orderings. For certain orderings, positive top-level superposition inferences *from* variables turn out to be redundant—a result that is relevant, surprisingly, in the context of equality elimination methods. The results are also extended to chaining calculi for non-symmetric transitive relations.

Keywords

Automated Theorem Proving, Basic Paramodulation, Chaining, Redundancy, Candidate Models, Counterexamples

Contents

1 Introduction

In 1988, Zhang & Kapur have proposed a clausal paramodulation calculus ZK. ZK is closely related to the superposition calculus S of Bachmair & Ganzinger (1990). ZK is weaker than S in that negative superposition also rewrites non-maximal terms of maximal negative equations. On the other hand, ZK is stronger than S as it comes without the equality factoring inference of the latter calculus. An inference derived from resolution with transitivity and ordered factoring, equality factoring not only represents an additional inference and increases the search space, it is also not very appealing from a conceptual point of view. In (Zhang & Kapur 1988), ZK was claimed to be refutationally complete, even in the presence of eager simplification of clauses by a variant of contextual rewriting that is sufficiently strong to eliminate all tautologies. However, Bachmair & Ganzinger (1990) have shown that tautologies cannot be eliminated without rendering ZK incomplete. Since then the question as to whether ZK is refutationally complete when tautologies are not eliminated, has remained open.

In (Bachmair & Ganzinger 1990, Bachmair & Ganzinger 1994*b*) it has also been shown that S can be combined with a variety of simplification techniques including those used in the Knuth/Bendix completion procedure. A general notion of redundancy based on logical entailment from smaller clauses has been established for justifying their admissibility. Accordingly, tautologies are always redundant and can be eliminated eagerly. Hence the proof techniques of these papers are not directly applicable to ZK. As a second problem, even if ZK should be refutationally complete, the calculus would be uninteresting from the practical point of view, if many of the standard techniques for eliminating redundancy (tautologies, subsumption, reduction) would not be compatible with the inference system.

This paper investigates these problems and gives answers which can be summarized as follows. We shall define a calculus SS, called strict superposition, which is superposition S without equality factoring. Therefore, SS is at the same time a substantially more restrictive version of ZK. We show that SS is refutationally complete. We also show that SS is compatible with a notion of redundancy by which most, but not all, of the major simplification techniques can be justified.

Our results are obtained by introducing a suitable notion of *direct rewrite proof* and modifying proof techniques based on candidate models and counterexamples in that we define these concepts, not in terms of semantic truth, but in terms of direct provability. Another contribution of this paper is that it makes the method of candidate models and counterexamples more transparent, separating its two central concepts, which are (i) the reduction of counterexamples (to certain candidate models) by inferences, and (ii) notions of redundancy which approximate the problem of identifying minimal such counterexamples. For SS the notion of counterexamples is weaker and, therefore, is accompanied by a weaker concept of redundancy. As weak counterexamples are not closed under reducing substitution parts of clauses, the lifting of strict superposition to non-ground clauses requires the technology for basic superposition (Bachmair, Ganzinger, Lynch & Snyder 1992, Nieuwenhuis & Rubio 1992*a*). This is another indication of why there are no obvious fixes for the gaps in the completeness proof for ZK that was attempted in (Zhang 1988).

We will also extend the results about strict superposition also to the case of ordered chaining calculi for general transitive relations. We will be able to show that transitivity (or composition) resolution inferences, the corresponding generalizations of the equality factoring inference, as they were required in (Bachmair & Ganzinger 1994*a*, Bachmair & Ganzinger 1994*c*, Bachmair & Ganzinger 1995) can also be dispensed with in a similar way.

2 Preliminaries

We assume the usual notions and notations about equational clauses, convergent term rewrite systems, and reduction orderings. We assume that equality, denoted by the symbol $\approx$, is the only predicate. Equations are also syntactically symmetric, that is, we do not distinguish between atoms $s \approx t$ and $t \approx s$. On the semantic side we consider *equality Herbrand interpretations* which are congruences on ground terms over the given signature. We will assume that equality interpretations are represented in the form of convergent ground rewrite systems I so that an atom $s \approx t$ is true in I if, and only if, $s \Downarrow_I t$, that is, if there exists a rewrite proof for $s \approx t$. When given a complete reduction ordering $\succ$ we may identify a ground equation with its ordered (according to $\succ$) counterpart, called rewrite rule. Given an interpretation I, we speak of a convergent ground rewrite system with respect to $\succ$ if the equations in I, when oriented with respect to $\succ$, represent a confluent rewrite system.

Superposition calculi are based on orderings which guide proof search and identify the normal forms of syntactic expressions which simplification is targeted at. A well-founded ordering $\succ$ on ground literals and ground terms is called admissible if (i) the restriction of $\succ$ to literals is a total ordering, (ii) the restriction of $\succ$ to ground terms is a total reduction ordering, and (iii) if L and L' are literals, then $L \succ L'$, whenever (iii.1) $\max(L) \succ \max(L')$, or (iii.2) $\max(L) = \max(L')$, and L is a negative, and L' is a positive literal. Hereby, $\max(L)$ denotes the maximal term (in $\succ$) of an equation or disequation L. If $\succ$ is an admissible ordering, its multi-set extension, again denoted by $\succ$, is called an admissible ordering on ground clauses. Admissible clause orderings are total and well-founded. The essence of condition (iii) is that replacing the maximal term in a literal by a smaller term yields a smaller literal in any admissible ordering. Condition (iii.2) makes clauses in which the maximal term occurs negatively larger than any clause with the same maximal term but with only positive occurrences of the latter. We observe that an admissible ordering on literals is firmly based on the ordering on terms. But there is some flexibility in extending a term ordering to literals. Provided (iii) is satisfied, any other syntactic criteria such as the ordering on the minimal terms or the division between skeleton and substitution positions (cf. below) may be employed freely to make the literal ordering become total.

By regarding literals as unit clauses we may use $\succ$ to also compare literals with clauses. In particular, a clause $C \vee s \approx t$ is called reductive for $s \Rightarrow t$ if $(s \approx t) \succ C$ and if $s \succ t$. Equivalent to an implication $\neg C \rightarrow s \approx t$, a reductive clause may be viewed as a conditional rewrite rule, rewriting s by t under the positive and negative conditions in $\neg C$. With reductive clauses, rewriting and recursive evaluation of the condition are terminating.

A clause ordering $\succ$ may be extended to sets of clauses by defining $N \succ M$ if, and only if, there exists a clause D in N such that, for all clauses D in M, $C \succ D$. If the clause ordering is well-founded, so is its extension to sets of clauses. In that case, the ordering may form the basis of definitions by noetherian induction. If N is a set of ground clauses and if C is a ground clause, by N_C we denote the set of clauses D in N with $C \succ D$. If C is in N then $N \succ N_C$. Subsequently all calculi will be parameterized by an admissible ordering.

3 Strict and Non-Strict Superposition

We assume for now that all clauses are ground. Problems related to lifting will be discussed in the Section 7 below.

The inference rules of strict (ground) superposition SS are given in the Figure 1. The ordering constraints restrict equational replacement such that only maximal terms of maximal equations in

Positive strict superposition

$$\frac{C \vee s \approx t \quad D \vee w[s] \approx v}{w[t] \approx v \vee C \vee D}$$

where (i) $s \succ t$, (ii) $w \succ v$, (iii) $(s \approx t) \succ C$, (iv) $(w \approx v) \succ D$, and (v) $(w \approx v) \succ (s \approx t)$.

Negative strict superposition

$$\frac{C \vee s \approx t \quad D \vee w[s] \not\approx v}{w[t] \not\approx v \vee C \vee D}$$

where (i) $s \succ t$, (ii) $w \succ v$, (iii) $(s \approx t) \succ C$, and (iv) $(w \not\approx v) \succeq \max(D)$.

Reflexivity resolution

$$\frac{s \not\approx s \vee C}{C}$$

where $(s \not\approx s) \succeq \max(C)$.

Ordered Factoring

$$\frac{C \vee s \approx t \vee s \approx t}{C \vee s \approx t}$$

where $(s \approx t) \succeq \max(C)$.

Figure 1: Strict superposition SS

---

the second premise (the negative premise) are replaced by the minimal term of a maximal equation of the first (the positive) premise. If D is a clause, by $\max(D)$ we denote the maximal literal in D.[1]

The calculus ZK (Zhang & Kapur 1988) is an early predecessor of SS. The ordering constraints in ZK restrict positive superposition for ground clauses in a similar way as in SS. ZK has constraints (i) and (ii), but not (v), with (iii) and (iv) slightly different due to a slightly different notion of literal orderings. In negative superposition also non-maximal terms of maximal equations have to be rewritten, that is, restriction (ii) is absent. ZK also has (unordered) factoring both for positive and for negative literals.

Bachmair & Ganzinger (1990) have shown that tautologies cannot be eliminated without rendering strict superposition incomplete. For example, the set of clauses

$$\begin{aligned} &\rightarrow \quad a \approx b, a \approx c \\ &\rightarrow \quad b \approx c \\ a \approx b, a \approx c \quad &\rightarrow \end{aligned}$$

[1] In order to simplify matters technically, in this paper we do not consider selection functions for negative literals. All results of the paper, however, can be extended appropriately.

where a, b, and c are constants, is unsatisfiable. However, if $a \succ b \succ c$, strict superposition cannot derive any other clause than the tautology $b \approx b,\ a \approx c \rightarrow a \approx c$.[2] One can see that even if unrestricted factoring and superposition into the smaller sides of negative equations is admitted one cannot derive the empty clause. Hence ZK, too, is not compatible with the elimination of tautologies. Since then the question as to whether SS is refutationally complete if one does not eliminate tautologies has remained open. The ordering constraints for SS do not admit superposition with tautologies as the positive premise, but superposition into tautologies is not excluded. In fact, in the example we may obtain a refutation as follows:

(1)	$\rightarrow a \approx b,\ a \approx c$	
(2)	$\rightarrow b \approx c$	
(3)	$a \approx b,\ a \approx c \rightarrow$	
(4)	$b \approx b,\ a \approx c \rightarrow a \approx c$	(1) into (3)
(5)	$b \approx b,\ b \approx c \rightarrow a \approx c,\ a \approx c$	(1) into (4)
(6)	$b \approx b,\ b \approx c \rightarrow a \approx c$	factoring (5)
(7)	$b \approx c,\ a \approx c,\ b \approx b,\ b \approx c \rightarrow$	(6) into (3)
(8)	$b \approx b,\ b \approx c,\ b \approx c,\ c \approx c,\ b \approx b,\ b \approx c \rightarrow$	(6) into (7)

From (8), superposition by (2) together with reflexivity resolution derives the empty clause.

It has since been shown (Bachmair & Ganzinger 1990), also for the basic variants of the calculus (Bachmair et al. 1992, Nieuwenhuis & Rubio 1992*a*), that superposition is complete, with all ordering constraints imposed, if one adds one more inference rule, called equality factoring:

Equality factoring

$$\frac{C \lor s \approx t \lor s \approx t'}{C \lor t \not\approx t' \lor s \approx t'}$$

where $(s \approx t) \succ (s \approx t')$, $s \succ t$, and $(s \approx t) \succ C$.

(Non-strict) superposition, denoted S, is defined as the extension of SS by equality factoring. Equality factoring is conceptually not very appealing as it combines a resolution inference into a certain instance of transitivity with factoring:

$$\cfrac{\cfrac{C \lor s \approx t \lor s \approx t' \quad (x \not\approx y \lor y \not\approx z \lor x \approx z)[t'/z]}{C \lor t \not\approx t' \lor s \approx t' \lor s \approx t'}}{C \lor t \not\approx t' \lor s \approx t'}$$

In other words, in addition to the superposition inferences, there are still some inferences with the transitivity clause required in S. As equality factoring only applies in the presence of disjunctions the differences between the strict and non-strict versions of superposition only become apparent for general, non-Horn clauses.

4 Candidate Models, Counterexamples, and Redundancy

A powerful method for proving the completeness of calculi for refutation proofs is centered around the concept of candidate models and the reduction of counterexamples. The main feature in our

[2]This example due to Chr. Lynch and W. Snyder is simpler than the one given in (Bachmair & Ganzinger 1990).

presentation of these concepts in the present paper is the separation between local aspects of reducing counterexamples (by inferences) and the global methods for identifying them. The latter will be captured by notions of redundancy: clauses are redundant if they can never become a minimal counterexample.

4.1 Basic Concepts

Let $\mathcal{I}$ be a (clausal) inference system, in which case we denote by $\mathcal{I}(N)$ the set of clauses that can be derived by applying an inference in $\mathcal{I}$ to premises in N. Suppose we have a mapping I, called model functor, assigning (equality) Herbrand interpretations I_N, called candidate models, to any given set N of ground clauses which does not contain the empty clause. Considering N, two cases may arise. I_N might be a model of N. Then one is finished as no contradiction can be derived from N, soundness of $\mathcal{I}$ assumed. Otherwise, there exists a clause C in N which is false in I_N. Such a clause C is called a counterexample (for I_N). If a counterexample exists, N must in particular contain a minimal counterexample with respect to the given clause ordering $\succ$. Suppose that whenever C is a minimal counterexample, there exists an inference in $\mathcal{I}$ from C and, possibly, additional premises in N, such that the conclusion D of the inference is smaller than C, i.e., $C \succ D$, and such that D is also a counterexample for I_N. If this property holds for all sets of clauses N we say that $\mathcal{I}$ has the reduction property for counterexamples (with respect to I). Inference systems that have the reduction property for counterexamples are refutationally complete:

Proposition 4.1 If $\mathcal{I}$ has the reduction property for counterexamples and if N is closed under $\mathcal{I}$, that is, $\mathcal{I}(N) \subseteq N$, either N contains the empty clause, or else N is satisfiable.

Proof. Suppose that $\mathcal{I}$ has the reduction property for counterexamples with respect to some model functor I. Any (minimal) counterexample C in N for I_N, by the reduction property, can be reduced to a smaller counterexample D which, if $\mathcal{I}(N) \subseteq N$, must be contained in N which is impossible. □

The proposition also shows that only inferences which reduce minimal counterexamples are essential. Clauses which can never become a minimal counterexample are, therefore, redundant. More formally, let us call a ground clause C (not necessarily in N) redundant with respect to a set N of ground clauses if there exist clauses $C_1, \dots, C_k$ in N such that $C_1, \dots, C_k \models C$ and $C \succ C_i$. Let $\mathcal{R}(N)$ denote the set of redundant clauses with respect to N, and let us call N saturated up to redundancy with respect to $\mathcal{I}$ whenever $\mathcal{I}(N \setminus \mathcal{R}(N)) \subseteq N \cup \mathcal{R}(N)$. If a clause C is redundant in N then it is not only entailed by N_C, but also by $N_C \setminus \mathcal{R}(N)$, the subset of non-redundant clauses in N smaller than C. Also note that any model of $N_C \setminus \mathcal{R}(N)$ is also a model of N. We observe that the redundancy criterion $\mathcal{R}$ and the inference system $\mathcal{I}$ are coupled through the common ordering they depend on.

Proposition 4.2 Let $\mathcal{I}$ be sound and have the reduction property, and let N be saturated up to redundancy with respect to $\mathcal{I}$. Then N is unsatisfiable if, and only if, N contains the empty clause.

Proof. Suppose that $\mathcal{I}$ has the reduction property for counterexamples with respect to some model functor I. Let M be an abbreviation for $N \setminus \mathcal{R}(N)$. Suppose that N does not contain the empty clause. If I_M is a model for M, it is also a model for N, and the proof is finished. Otherwise M contains a minimal counterexample D for I_M. As $\mathcal{I}$ has the reduction property we may find clauses

C_i in M such that some inference from premises D and C_i has a conclusion D', where $D' \prec D$ and D' is a counterexample for I_M. By the saturation assumption, D' is an element of $N \cup \mathcal{R}(N)$ since the premises of the inference are all in $N \setminus \mathcal{R}(N)$. If D' is in $\mathcal{R}(N)$ then it is entailed by $N_C \setminus \mathcal{R}(N)$ which is a subset of M. This contradicts the assumption that D is the minimal counterexample to I_M in M. If D' is not in $\mathcal{R}(N)$ it must be in M, which again is a contradiction. We have shown that M does not contain a counterexample for I_M, therefore, I_M is a model for M and, hence, also a model for N. □

Our previous papers, e.g., (Bachmair & Ganzinger 1994*b*), define formal notions of theorem proving processes, as sequences of steps of deduction and deletion of redundant clauses. The stability of redundancy under deduction and deletion (of redundant clauses), is shown. Under certain assumptions about fairness any such process is refutationally complete. In particular, redundant clauses can be eliminated eagerly in any theorem proving process. All this is essential for effectively and efficiently constructing saturated sets of clauses, but orthogonal to the topic of the present paper.

4.2 Constructing Candidate Models

The previous propositions leave us with the problem, given $\mathcal{I}$, how to construct a model functor I such that $\mathcal{I}$ has the reduction property for counterexamples with respect to I. In the sequel we present a construction that can be applied to many clausal inference systems for equality and other transitive relations. Recall that equality interpretations are given in the form of a convergent (with respect to $\succ$) ground rewrite system R, such that an equation $s \approx t$ is true in R if and only if $s \Downarrow_R r$.

I_N will consist of equations that are produced by certain reductive clauses in N. Let R be an interpretation. A clause C of the form $D \vee s \approx t$ is called productive on R if

(i) C is reductive for $s \Rightarrow t$

(ii) D is a counterexample for $R \cup \{s \approx t\}$, and

(iii) s is irreducible by R.

We also say that C produces $s \approx t$ from R. With this, we inductively define I_N as

$$I_N = \{s \approx t \mid \exists C \in N : C \text{ produces } s \approx t \text{ from } I_{N_C}\}.$$

(As $N \succ N_C$ whenever C is in N, the I_{N_C} can assumed to be well-defined by induction hypothesis.) I_N is the set of equations produced by those clauses C in N which are productive on the partial interpretations I_{N_C}. In that case, C is a counterexample for I_{N_C} and, as a result of producing $s \approx t$, becomes true in $I_{N_C} \cup \{s \approx t\}$. However, its side literals D (assuming that $C = D \vee s \approx t$) remain a counterexample (that is, the condition $\neg D$ of the conditional rewrite rule $\neg D \rightarrow s \approx t$ remains true) when $s \approx t$ is added to I_{N_C}, cf. condition (ii). The mapping I is monotone in that whenever C and D are two clauses with $C \preceq D$, then $I_{N_C} \subseteq I_{N_D} \subseteq I_N$. Whenever the set N is known from the context or assumed to be arbitrary then we will abbreviate I_N and I_{N_C} by, respectively, I and I_C.

By the condition (iii), the sets I_C and I are left-reduced rewrite systems with respect to $\succ$ and, therefore, convergent. The technicalities of the clause ordering and the convergence of the rewrite systems imply that if D is a productive clause in N, and if C, D', and D'' are other clauses such that $D'' \succ D' \succ D \succeq C$ then C is a counterexample for $I_{D'}$ if, and only if, C is a counterexample for $I_{D''}$.

Let us explain this construction by means of an example. Suppose we have constants $a \succ b \succ c \succ d$. The following table lists a set of clauses in ascending order from bottom to top, together with the rewrite rules that are produced into the interpretation by that clause.

C			I	remarks
$a \approx b$	$\rightarrow$	$a \approx d$		true in I_C
$a \approx d$	$\rightarrow$	$b \approx c$		false in I_C and I
$c \approx d$	$\rightarrow$	$a \approx c,\ b \approx c$	$a \approx c$	reductive and false in I_C
	$\rightarrow$	$c \approx d$	$c \approx d$	

In the end, $I = \{c \approx d,\ a \approx c\}$ such that all but the second clause are true in the candidate model. We observe that a superposition inference

$$\frac{c \approx d \rightarrow a \approx c,\ b \approx c \quad a \approx d \rightarrow b \approx c}{c \approx d,\ c \approx d \rightarrow b \approx c,\ b \approx c}$$

exists between the third and the second clause which produces a smaller counterexample $c \approx d,\ c \approx d \rightarrow b \approx c,\ b \approx c$ for I. Indeed, if a clause C such as $a \approx d \rightarrow b \approx c$ is false in I_C, its negative literal $a \approx d$ converges in I_C. Consequently, its maximal term a must be reducible by a rule in I_C. As this rule is produced by a clause D in N_C, a reducing superposition inference exists. From the condition (i) in the model functor construction, the side literals in D are false, such that the reduced clause is indeed a counterexample.

Theorem 4.3 Let N be a set of ground clauses not containing the empty clause. Let C be the minimal counterexample in N for I_N. Then there exists an inference in S from C such that

(i) its conclusion is a counterexample for I_N and is smaller than C; and

(ii) if the inference is a superposition inference then C is its negative premise and the positive premise is a productive clause.

We shall see the details of the required reasoning in the closely related proof for the Theorem 7.2 below.

The theorem in part (i) states that S has the reduction property, and hence is refutationally complete and compatible with the removal of redundant clauses. In addition, part (ii) gives further semantic restrictions to S which may be exploited with a notion of redundancy for inferences in addition to the one for clauses. We shall return to this in the Section 7 when we will deal with lifting.

To see what goes wrong for strict superposition, let us now apply the construction to the critical example of the Section 3.

clauses C			I	remarks
$a \approx b,\ a \approx c$	$\rightarrow$			true
	$\rightarrow$	$a \approx b,\ a \approx c$		minimal counterexample
	$\rightarrow$	$b \approx c$	$b \approx c$	

The clause $D = a \approx b,\ a \approx c$ is non-productive, as producing $a \approx b$ would make $a \approx c$ become true, contradicting the condition (i). In the absence of equality factoring, no inference can reduce this counterexample. (Equality factoring can, as it derives $b \approx c \rightarrow a \approx c$ from D.)

We have shown in the Section 3 that a contradiction can be derived if one computes inferences with the tautology. If there is at all an extension of the concept of model candidates and counterexample reduction to cover this situation, it must come with a more liberal notion of counterexamples where certain tautologies, although semantically true, can be considered "weak" counterexamples for some interpretations. Such a concept must be proof-theoretic in nature, and will be developed next.

5 Weak Counterexamples

Let R be a ground rewrite system. An equation $s \approx t$ is said to have a direct rewrite proof in R if, and only if, $s \Downarrow_{R_{s,t}} t$, where $R_{s,t}$ denotes the set of equations $l \approx r$ in R such that $s \approx t \succeq l \approx r$ in the given literal ordering. In a direct rewrite proof only those rewrite rules may be applied that are not larger than the equation to be proved. We call a ground clause C a weak counterexample for R if, and only if, (i) for each negative literal $u \not\approx v$ in C we have $u \Downarrow_R v$ and, (ii) no positive equation in C has a direct rewrite proof in R. Clearly, any counterexample is also a weak counterexample, but the converse is not true in general.

Given an inference system $\mathcal{I}$ and a model functor I we say that $\mathcal{I}$ has the reduction property for weak counterexamples (with respect to I) if whenever a (nonempty) clause C in N is the minimal weak counterexample for I_N then there exists an inference in $\mathcal{I}$ from C and, possibly, additional premises in N, such that the conclusion D of the inference is smaller than C, i.e., $C \succ D$, and such that D is also a weak counterexample for in I_N.

Proposition 5.1 If $\mathcal{I}$ has the reduction property for weak counterexamples and if N is closed under $\mathcal{I}$, that is, $\mathcal{I}(N) \subseteq N$, either N contains the empty clause or else N is satisfiable.

Again, identifying minimal weak counterexamples leads to a compatible notion of redundancy. Call a ground clause C (not necessarily in N) (weakly) redundant with respect to a set N of ground clauses if for any convergent ground rewrite system R (with respect to $\succ$) for which C is a weak counterexample there exists a weak counterexample D in N for R such that $C \succ D$. Let $\mathcal{R}^w(N)$ denote the set of redundant clauses (based on weak counterexamples) with respect to N, and let us call N saturated up to weak redundancy with respect to $\mathcal{I}$ whenever $\mathcal{I}(N \setminus \mathcal{R}^w(N)) \subseteq N \cup \mathcal{R}^w(N)$. Note that if R is an interpretation such that no clause in $N \setminus \mathcal{R}^w(N)$ is a weak counterexample for R then also no clause in N can be a weak counterexample for R, and, in particular, R is a model for N. Conversely, if C in $\mathcal{R}^w(N)$ is a weak counterexample for R then $N \setminus \mathcal{R}^w(N)$ also contains a (smaller) weak counterexample for R.

Proposition 5.2 Let $\mathcal{I}$ be sound and have the reduction property for weak counterexamples, and let N be saturated up to weak redundancy with respect to $\mathcal{I}$. Then N is unsatisfiable if and only if N contains the empty clause.

In applying these concepts to strict superposition, we modify the previous definitions of productive clauses and of the model functor I_N, in that we replace "counterexample" by "weak counterexample". Let I^w_N denote the modified constructions, again omitting the index N when no confusion can arise, and writing I^w_C for $I^w_{N_C}$ when referring to the "partial" interpretations for the subsets N_C of N. The condition (ii) for productive clauses C now simply reduces to requiring that C be a weak counterexample for I_C. If $C = D \vee s \approx t$ is a weak counterexample for I^w_C, no positive literal L in D can have a direct rewrite proof in $I^w_C \cup \{s \approx t\}$'. In fact, as the clause has to be reductive for $s \approx t$, we have $(s \approx t) \succ L$ so that $s \approx t$ cannot be applied in any direct rewrite proof of L.

In the critical example we now obtain this candidate model:

C			I^w	remarks
$a \approx b,\ a \approx c$	$\rightarrow$			minimal weak counterexample
	$\rightarrow$	$a \approx b,\ a \approx c$	$a \approx b$	
	$\rightarrow$	$b \approx c$	$b \approx c$	

Now the clause $a \approx b,\ a \approx c$ is productive. Superposing this clause one may reduce the counterexample to $b \approx b,\ a \approx c \rightarrow a \approx c$. If we analyse this tautology with respect to the rewrite system

$\{a \approx b,\ b \approx c\}$ we find out that it constitutes a weak counterexample. In fact, strict superposition does have the reduction property for weak counterexamples and, hence, is refutationally complete for ground clauses.

Theorem 5.3 Let N be a set of ground clauses not containing the empty clause. Let C be the minimal weak counterexample in N for I_N^w. Then there exists an inference in SS from C such that

(i) its conclusion is a weak counterexample for I_N^w and is smaller than C; and

(ii) if the inference is a superposition inference (left or right) then C is its negative premise and the positive premise is a productive clause.

For the proof idea we again refer to the closely related proof of the Theorem 7.2.

The theorem in part (i) states that SS has the weak reduction property for I^w, and hence is refutationally complete and compatible with the removal of weakly redundant clauses. As for S, part (ii) gives further semantic restrictions to SS which may be exploited with an additional notion of redundancy for inferences.

6 Simplification Afforded by Weak Redundancy

We will briefly investigate the principle simplification methods in superposition theorem proving with regard to their admissibility based on weak redundancy. We treat the ground case. For the non-ground case additional consideration have to be taken into account which arise from lifting with the basic strategy, cf. Section 7. In general, simplification is a derivation step (on sets of clauses) of the form

Simplification:

$$N \cup \{C\} \vdash N \cup \{D\}$$

such that C and D are logically equivalent, $C \succ D$, and C is in $\mathcal{R}^w(N \cup \{D\})$.

In other words, by a simplification of C to D we mean that adding D is sound and causes C to be redundant. Reduction is an admissible simplification:

Positive reduction:

$N \cup \{s \approx t,\ C \vee w[s] \approx u\} \vdash N \cup \{s \approx t,\ C \vee w[t] \approx u\}$, if $(w[s] \approx u) \succeq (s \approx t)$ and $s \succ t$.

Soundness is trivial. We show that $C \vee w[s] \approx u$ is redundant in $\{s \approx t,\ C \vee w[t] \approx u\}$. Let R be a convergent ground rewrite system with respect to $\succ$ such that neither $s \approx t$ nor $C \vee w[t] \approx u$ is a weak counterexample. We argue that, then, $C \vee w[s] \approx u$ is not a weak counterexample for R either. Suppose that $C \vee w[t] \approx u$ is not a weak counterexample because of the existence of a direct rewrite proof for $w[t] \approx u$ in R. As R is convergent, this rewrite proof may be combined with the direct rewrite proof for $s \approx t$ to a rewrite proof for $w[s] \approx u$. Since $(w[s] \approx u) \succeq (s \approx t)$, any rule used in this proof is not greater than $w[s] \approx u$, hence the rewrite proof is direct.

Reduction of negative literals is less restricted as there is no difference between regular and direct rewrite proofs for negative literals.

Negative reduction:

$N \cup \{s \approx t,\ C \vee w[s] \not\approx u\} \vdash N \cup \{s \approx t,\ C \vee w[t] \not\approx u\}$, if $s \succ t$.

Tautology elimination must be weaker than usual. By a direct tautology we mean a clause C such that for any convergent rewrite system R with respect to $\succ$ for which $s \Downarrow_R t$, for all negative literals $s \not\approx t$ in C, there exists a positive literal in C which has a direct rewrite proof in R. Direct tautologies can be eliminated.

Direct tautology elimination:
$N \cup \{C\} \vdash N \quad$ if C is a direct tautology.

Here is a sufficient, decidable criterion for direct tautologies:

PROPOSITION 6.1 *C is a direct tautology if C is of the form $C_n \vee C_p \vee w[s] \approx t$, with C_n the subclause of negative literals in C, the context $w[\_]$ nonempty, and $w[s] \succeq t$, such that $C_n \vee s \approx u$ and $C_n \vee w[u] \approx t$, for some $u \prec s$, are tautologies.*

Proof. We show that under the given assumptions, C is a direct tautology. The case is trivial if $w[s] = t$. Let us, therefore, assume that $w[s] \succ t$. Let R be a convergent ground rewrite system for which the negative equations C_n converge. As $C_n \vee s \approx u$ and $C_n \vee w[u] \approx t$ are tautologies, the equations $s \approx u$ and $w[u] \approx t$ have (possibly non-direct) rewrite proofs in R. The context $w[\_]$ is nonempty, and therefore, $w[s] \succ s$ in any reduction ordering. Placing the rewrite proof for $s \approx u$ into the context $w[\_]$, therefore, provides us with a direct rewrite proof of $w[s] \approx w[u]$. Since R is convergent, the latter may be combined with the rewrite proof for $w[u] \approx t$ into a direct rewrite proof of $w[s] \approx t$. □

In practice, one might implement the tautology test provided by the proposition by ground completing C_n, and then proving $w \approx t$ with a rewrite proof in which the left side of any appearing rewrite rule is smaller than w.

Strict subsumption is as usual since being a weak counterexample requires all literals of the clause to be weak counterexamples.

Subsumption:
$N \cup \{C,\ C \vee D\} \vdash N \cup \{C\}$

To summarize, we have shown that with certain additional restrictions, the usual simplification techniques for non-strict superposition are also admissible for the strict calculus.

7 Lifting

7.1 The Lifting Problem for S and SS

When lifting superposition inferences to non-ground clauses one wants to exclude superposition into terms that occur at or below a variable in the respective ground instances. Otherwise, unification is not an effective filter for the inference, and superposition would be extremely prolific. The restricted superposition system no longer has the reduction property . In fact, ground superposition systems, if target terms for replacement are arbitrarily restricted, are obviously incomplete. Fortunately, sets of ground clauses N which are obtained by forming all ground instances of a given set of non-ground clauses M have the crucial property that if $C\sigma$, for C in M, is a clause in N, so is any clause $C\tau$ for which τ can be obtained by reducing the substitution σ with respect to any rewrite system I, in particular, with respect to I_N.

Let us explain the situation using an example. Consider the inference

$$\frac{D \vee s \approx t \quad (C[x] \vee x \approx u)[s/x]}{(D \vee C[x] \vee t \approx u)[s/x]}$$

and, for simplicity, assume that all expressions, except for the ones containing the variable x, are ground. That is, we deal with a set of clauses N containing $s \approx t$ and all the ground instances of $C[x] \vee x \approx u$, but disallow superposition into subterms at x. For the inference to be ordered, we have $s \succ t$. From part (ii) of the Theorem 4.3 we observe that we may assume the positive premise of the inference to produce $s \approx t$ into I_N. Then, however, the negative premise cannot be the minimal counterexample. For if $(C[x] \vee x \approx u)[s/x]$ is false in I_N then, as $s \approx t$ is true, the reduced instance $(C[x] \vee x \approx u)[t/x]$, also an element of N, is false in I_N and smaller. We conclude that the non-ground version of S, when restricted to subterms above variables, has the reduction property for counterexamples in schematic sets of ground clauses, by which we mean sets which consist of all the ground instances of some set of non-ground clauses. From this refutational completeness for general clauses follows.

In the case of strict superposition with "counterexample" replaced by "weak counterexample" the situation is more involved although we have the analogous Theorem 5.3. Consider the example in the Figure 2 where we assume $a \succ b \succ c \succ d \succ e$ and consider the instances of a non-ground clause $D = x \approx b,\ x \approx e$ along with some other ground clauses and again list the clauses in ascending order (from bottom to top), side-by-side with the equations they generate in the model construction I^w based on weak counterexamples. One observes that the first clause is true in I^w but it is a

clauses C	I^w	remarks
$\rightarrow\ \boxed{a} \approx b,\ \boxed{a} \approx e$		$D[a/x]$
$\rightarrow\ a \approx c,\ a \approx d$	$a \approx c$	
$\rightarrow\ b \approx b,\ b \approx e$		$D[b/x]$
$\rightarrow\ c \approx b,\ c \approx e$		$D[c/x]$
$\rightarrow\ d \approx b,\ d \approx e$		$D[d/x]$
$\rightarrow\ e \approx b,\ e \approx e$		$D[e/x]$
$\rightarrow\ c \approx e$	$c \approx e$	
$\rightarrow\ d \approx e$	$d \approx e$	

Figure 2: Reduction of Weak Counterexamples

(minimal) weak counterexample to I^w. If one forbids superposition into the (boxed) substitution position a of the first clause, no inference is possible so that the weak counterexample cannot be reduced; the correspondingly reduced instance (the fourth clause) of $x \approx b,\ x \approx e$ is present, but is not a *weak* counterexample. In short, SS does generally not have the reduction property for weak counterexamples in schematic sets of clauses.

A remedy to this problem will be to consider clause sets N which only contain "reduced" (with respect to I^w_N) instances of the given non-ground clauses. The precise definition of "reduced" will be closely related to the reductions that are admitted in direct rewrite proofs. The same technique was needed for proving the refutational completeness of basic superposition (Bachmair & Ganzinger 1990, Bachmair et al. 1992, Nieuwenhuis & Rubio 1992*a*) and is known to lead to a somewhat more complex notion of redundancy. This indicates that a non-basic version of strict superposition might not be very interesting from a practical point of view, motivating us to generalize our results to the basic variant of strict superposition.

7.2 Closures

Basic strategies are best formulated with constrained clauses (Nieuwenhuis & Rubio 1992*b*). In this paper we do not want to describe details of how to actually compute with constrained clauses but only want to treat matters on a level such that the lifting problem can be considered as theoretically solved. Therefore we just introduce as much additional syntax as is needed for marking those subterms in ground clauses in which replacement is not allowed. Specifically, we assume that the positions in any ground clause are marked according to whether or not they belong to the substitution by which they were obtained as instance of a non-ground clause. Generally for any ground expression (term, literal, clause) we assume that their substitution positions be marked. We shall use the same meta-variables as before (C and D for clauses, L for literals, and s, t, u, v, w for terms) for the respective kinds of marked ground expressions.

When we need to explicitly represent the markings, we shall use a representation by closures. A (ground) closure is a pair $C \cdot \sigma$ consisting of a (general) clause C and a ground substitution σ such that $C\sigma$ is a ground clause. Then the substitution positions in the instance $C\sigma$ are exactly the ones at or below a variable in C. We identify any two closures $C \cdot \sigma$ and $D \cdot \tau$ whenever they represent the same ground clause with the same markings. That is the case precisely when $C\sigma = D\tau$ and when the skeletons C and D are identical up to variable names, that is, $C\rho = D\rho$, for some variable renaming substitution ρ. (Subsequently we shall use the notation $E \equiv E'$ whenever two syntactic expressions E and E' only differ in the names of their variables.) In particular, we may assume that the variables in the skeleton C of a closure $C \cdot \sigma$ are always freshly chosen to achieve variable disjointness where required. Also, any two closures can be brought into the form $C \cdot \sigma$ and $D \cdot \sigma$, respectively, with a common substitution σ.

Admissible orderings are defined as as before. With the extended syntax, the literal ordering may now also depend on the marking of positions in clauses whenever the maximal terms and the polarity of two literals under comparison are the same (cf. Section 7.4).

A notion central to the lifting of basic inference systems is that of a reduced clause. For reduced clauses no inferences at substitution positions need to be considered. Given a rewrite system R, in reduced clauses, all substitution positions in a literal L are reduced by rules in R which are smaller than L. More precisely, we say that a literal $L[s]_p$ is order-reducible (at position p) by an equation $s \approx t$, if $s \succ t$ and $L \succ (s \approx t)$. In any admissible ordering, a negative literal is order-reducible by $s \approx t$ if and only if it is reducible by $s \approx t$. On the other hand, a positive equation $u \approx v$ is order-reducible by $s \approx t$ only if $(u \approx v) \succ (s \approx t)$ in the literal ordering. The restriction for order reducibility is slightly stronger than the one for direct rewrite proofs where only $(u \approx v) \succeq (s \approx t)$ is required for the application of $s \approx t$ in $u \approx v$. A literal is order-reducible by R if it is order-reducible by some equation in R. Likewise, a clause is called order-[ir]reducible at p if the literal to which p belongs is order-[ir]reducible at p. “Order-irreducible” is the same as “not order-reducible.” A ground clause is simply called reduced with respect to R if it is order-irreducible with respect to R at all substitution positions.

For concrete examples of order-reducible, respectively, order-irreducible clauses in the context of a specific literal ordering the reader is referred to the Section 7.4.

Positive strict basic superposition

$$\frac{(C \lor s \approx t) \cdot \sigma \quad (D \lor w[s'] \approx v) \cdot \sigma}{(w[t] \approx v \lor C \lor D) \cdot \sigma}$$

where $s\sigma = s'\sigma$, and (i) $s\sigma \succ t\sigma$, (ii) $w\sigma \succ v\sigma$, (iii) $(s \approx t) \cdot \sigma \succ C \cdot \sigma$, (iv) $(w \approx v) \cdot \sigma \succ D \cdot \sigma$, and $(w \approx v) \cdot \sigma \succ (s \approx t) \cdot \sigma$, and (v) s' is not a variable

Negative strict basic superposition

$$\frac{(C \lor s \approx t) \cdot \sigma \quad (D \lor w[s'] \not\approx v) \cdot \sigma}{(w[t] \not\approx v \lor C \lor D) \cdot \sigma}$$

where $s\sigma = s'\sigma$, and (i) $s\sigma \succ t\sigma$, (ii) $w\sigma \succ v\sigma$, (iii) $(s \approx t) \cdot \sigma \succ C \cdot \sigma$, (iv) $(w \not\approx v) \cdot \sigma \succeq \max(D \cdot \sigma)$, (v) s' is not a variable.

Reflexivity resolution

$$\frac{(s \not\approx s' \lor C) \cdot \sigma}{C \cdot \sigma}$$

where $s\sigma = s'\sigma$ and $(s \not\approx s') \cdot \sigma \succeq \max(C \cdot \sigma)$.

Ordered Factoring

$$\frac{(C \lor s \approx t \lor s' \approx t') \cdot \sigma}{(C \lor s \approx t) \cdot \sigma}$$

where (i) $s\sigma = s'\sigma$, $t\sigma = t'\sigma$, $(s \approx t) \cdot \sigma \succeq \max(C \cdot \sigma)$, and (ii) $s \equiv s'$ and $t \equiv t'$.

Figure 3: Strict basic superposition SBS

7.3 Strict Basic Superposition

The inference rules of (ground) strict basic superposition SBS are given in the Figure 3. One observes that the calculus is simply the strict superposition calculus applied to marked ground clauses represented by closures such that, with the condition (v), term replacement is not allowed at substitution positions. Another characteristic property of the basic variant is that when generating a clause by an inference, all its skeleton positions correspond to skeleton positions in the premises so that no new skeleton positions are ever generated. Observe that the condition (ii) in the factoring inference requires that also the skeletons of the factor equations to be identical (up to variable renaming).

In investigating the reduction property for SBS we assume that candidate models I^w for sets of ground clauses N be defined as before. Produced equations keep their marking of substitution positions as present in the producing clause.

A set of ground clauses N is said to be reduced if any clause C in N is reduced by I^w_C. From the monotonicity of I^w, and considering the details of the clause orderings, we observe that if C is reduced by I^w_C it is also reduced by I^w_N. No clause greater than or equal to C can produce a rule that is smaller than some literal in C. We say that a reduced subset M of N is maximal whenever it contains all clauses in N which are reduced with respect to I^w_M.

PROPOSITION 7.1 Any set of clauses contains a maximal reduced subset.

Proof. Define $M = M(N)$ as the set of all clauses C in N which are reduced with respect to $I^w_{M(N_C)}$, where $M(N_C)$ denotes, by induction hypothesis, the maximal reduced subset of N_C. Observe that $M(N_C) = M_C$, for any C. Moreover, a clause C in N is reduced with respect to $I^w_{M(N_C)}$ and, hence, with respect to $I^w_{M_C}$, if and only if C is reduced with respect to I^w_M. (Productive clauses in $M \setminus M_C$ produce equations which are maximal in a clause not smaller than C. Such equations cannot be smaller than any literal in C, hence, cannot order-reduce redexes in C.) We conclude that M is reduced and maximal. □

THEOREM 7.2 Let N be a reduced set of clauses not containing the empty clause. Let C be the minimal weak counterexample in N for I^w_N. Then there exists an inference in SBS from C such that

(i) its conclusion D is a weak counterexample for I^w_N, is smaller than C, and is reduced with respect to I^w_N; and

(ii) if the inference is a superposition inference then C is its negative premise, and the positive premise is a productive clause.

Proof. The proof proceeds by a case analysis of the various kinds of counterexamples that might exist for I^w_N. Suppose the minimal weak counterexample C is of the form $C' \vee w \approx v$ where C is reductive for $w \Rightarrow v$, hence in particular $w \succ v$. As C is a weak counterexample, C' is a weak counterexample for I^w_N, and C is not productive. If a clause of this form is not productive, its maximal term w must be reducible by a rule $s \approx t$ in I^w_C which is produced by a clause $D'' = D' \vee s \approx t$ smaller than C. It follows that $(w \approx v) \succ (s \approx t)$. $w \approx v$ is order-irreducible by I^w_N, hence irreducible by $s \approx t$ at all its substitution positions. Therefore, s cannot occur at a substitution position in w, so that

$$\frac{D' \vee s \approx t \quad C' \vee w[s] \approx v}{w[t] \approx v \vee C' \vee D'}$$

is an inference in SBS, assuming that the conclusion $D = w[t] \approx v \vee C' \vee D'$ inherits the marking of the substitution positions from its premises as specified in SBS. From what we have said before, the ordering constraints for the inference are, in fact, satisfied. As the maximal term $w[s]$ in the negative premise is reduced to $w[t]$, we have $(w[s] \approx v) \succ (w[t] \approx v)$ in any admissible ordering. From $(w[s] \approx v) \succ s \approx t \succ D'$ we, therefore, infer that $C \succ D$. As D'' is productive, D' is a weak counterexample for I^w_N. C is a weak counterexample by assumption, and, hence, so are C' and $C' \vee D'$. $s \approx t$ is true in I^w_N, so that if $w[t] \approx v$ had a direct rewrite proof in I^w_N, by applying $s \approx t$ to $w[s]$, the proof could be extended to a direct rewrite proof of $w \approx v$, and C would not have been be a weak counterexample. In short, D is a weak counterexample.

It remains to be shown that D is reduced. The $C' \vee D'$ part of D inherits the substitution positions from the respective premises which are order-irreducible by assumption. It is also easy to see that the new literal $w[t] \approx v$ is order-irreducible. If a redex occurs at a substitution position in t, reducing it by any rewrite rule would order-reduce the corresponding position in $s \approx t$. Redexes at substitution positions outside t in $w[t] \approx v$ also occur in the substitution part of $w[s] \approx v$ and are, by assumption, irreducible by rules smaller than $w[s] \approx v$. In summary, D is reduced.

If the minimal weak counterexample C is of the form $w \not\approx v \vee C'$ where $(w \not\approx v) \succeq \max(C')$ then two cases may arise. Suppose that $w \succ v$. For C to be a weak counterexample, $w \approx v$ has a rewrite proof in I_N^w, so that w is reducible by a rule $s \approx t$ in I_C^w. Suppose that $s \approx t$ is produced by $D'' = D' \vee s \approx t$. Then the proof for this case is essentially as before, deriving a smaller reduced weak counterexample by a negative strict basic superposition inference from D'' and C. If $w = v$ we may use reflexivity resolution to construct a smaller reduced weak counterexample.

The remaining case for C is that of a (non-reductive, hence, non-productive) clause of the form $(C' \vee s \approx t \vee s' \approx t') \cdot \sigma$, with $(s \approx t) \cdot \sigma \succeq \max(C)$, $s\sigma \succ t\sigma$. Moreover, since neither is greater than the other in the literal ordering, $(s \approx t) \cdot \sigma$ and $(s' \approx t') \cdot \sigma$ are identical up to variable renaming. In particular, $s\sigma = s'\sigma$, $t\sigma = t'\sigma$, $s \equiv s'$ and $t \equiv t'$. This form of a weak counterexample can be reduced by a factoring inference. □

The theorem in part (i) states that SBS has the reduction property for weak counterexamples in reduced sets of clauses. It in addition asserts that the smaller counterexample is again a reduced clause, which is important for effective saturation.

A set of clauses N is called schematic, whenever there exists a set of (general) clauses M such that N is the set of closures $C \cdot \sigma$ with C in M and σ an arbitrary ground substitution. The reduction property of SBS implies its refutational completeness for schematic sets of clauses. We will, however, right away prove a stronger result which shows that SBS is, in addition, compatible with an appropriate notion of redundancy. The new redundancy criterion will be the same as $\mathcal{R}^w$ except that it refers only to *reduced* weak counterexamples. Let us call a clause C redundant with respect to a set of clauses N if for all convergent (with respect to $\succ$) rewrite systems R for which C is a weak counterexample and reduced, N contains a reduced weak counterexample D for R such that $C \succ D$. By $\mathcal{R}^c(N)$ we denote the set of clauses redundant with respect to N. N is called saturated up to redundancy with respect to SBS if $\mathsf{SBS}(N \setminus \mathcal{R}^c(N)) \subseteq N \cup \mathcal{R}^c(N)$.

In order to show that $N \cup \{C\} \vdash N \cup \{D\}$ is a simplification that is compatible with $\mathcal{R}^c$, one now has to show that C and D are logically equivalent, $C \succ D$, and C is in $\mathcal{R}^c(N \cup \{D\})$. A sufficient criterion ensuring that C is in $\mathcal{R}^c(N \cup \{D\})$ is when C is in $\mathcal{R}^w(N \cup \{D\})$ and when, for any convergent rewrite system, D is reduced whenever C is reduced. The latter will be the case if subterms s at substitution position in a literal L of D also occur in C at a substitution position of a literal L' such that $L' \succeq L$. With this criterion of relative order reducibility, the techniques discussed in the Section 6 can be appropriately refined. The specificities of $\mathcal{R}^c$ with regard to reducibility provide us with additional simplification rules. The following rule has turned out to be of some use in the context of equality elimination transformations (Bachmair, Ganzinger & Voronkov 1997):

Ground reflexivity resolution:

$$N \cup \{(C \vee x \not\approx y) \cdot \sigma\} \;\vdash\; \begin{cases} N \cup \{C \cdot \sigma\}, & \text{if } x\sigma = y\sigma \\ N, & \text{if } x\sigma \neq y\sigma \end{cases}$$

where x, y are variables.

Because of the requirement that x and y be variables, relative order reducibility can easily be shown. In the first variant, the simplified clause is clearly equivalent and smaller. In the second case, suppose that R is a convergent rewrite system and $(C \vee x \not\approx y) \cdot \sigma$ is a weak counterexample for R. Then $x\sigma \approx y\sigma$ must converge in R. As these two terms are different, at least one of them must be reducible by R, hence $x \not\approx y \cdot \sigma$ is order-reducible by R. Hence, whenever $(C \vee x \not\approx y) \cdot \sigma$ is order-irreducible, it is not a weak counterexample. Therefore, $(C \vee x \not\approx y) \cdot \sigma$ is redundant.

On the level of non-ground closures $C \cdot \gamma$, with equality constraints γ generalizing the concept of ground subsitutions in ground closures, the simplification rule can be represented as

Reflexivity resolution:
$N \cup \{(C \vee x \not\approx y) \cdot \gamma\} \vdash N \cup \{C \cdot (\gamma \wedge x = y)\}$, where x, y are variables.

THEOREM 7.3 Let N be a set of clauses which is saturated up to redundancy with respect to SBS. Moreover, assume that N contains a schematic subset K such that every clause in $N \setminus K$ is a logical consequence of K. Then either N contains the empty clause, or else N is satisfiable.

Proof. Let N not contain the empty clause, let N' denote $N \setminus \mathcal{R}^c(N)$, and let M' be a maximal, reduced subset of N'. Assume for the purpose of deriving a contradiction that M' contains a minimal weak counterexample C for $I_{M'}$. As C is not redundant with respect to N it is also non-redundant with respect to its subset M'. Applying the Theorem 7.2, part (i), to M', we infer the existence of a smaller, reduced weak counterexample D for $I_{M'}$. As N is saturated, D is an element of $N \cup \mathcal{R}^c(N)$. By definition of $\mathcal{R}^c$ this implies the existence of a clause D' in N which is smaller than C, reduced with respect to $I_{M'}$, and a weak counterexample for $I_{M'}$. The smallest such D' cannot be redundant in N, hence it is in N' and even in M', since it is reduced. But this is a contradiction to the minimality of C. Therefore, $I_{M'}$ is a model of M'.

We next show that $I_{M'}$ is also a model for K. If C is any clause in K, then we may consider the clause D in M' which results from C normalizing any subterm at a substitution position by $I_{M'}$. If D is in N' then, by maximality of M', it is also contained in M' and therefore true in $I_{M'}$, which in turn implies that C is true in $I_{M'}$. If D is redundant with respect to N then, if it were counterexample for $I_{M'}$, N would also contain a non-redundant, reduced weak counterexample for $I_{M'}$, which we have already shown to be impossible. Again, D, hence C, is true in $I_{M'}$. In summary, all clauses in K, and therefore all clauses in N, are true in $I_{M'}$. □

Whenever N is the closure under SBS (ignoring inferences involving redundant clauses) of some initially given set N_0 of (general) clauses then the set K of ground instances of N_0 is schematic and, hence, this theorem can be applied.

In summary, SBS can be lifted to any suitable notion of constrained clauses.

7.4 Optimized Variable Chaining

The preceding completeness results have been proved for the class admissible orderings. We shall now define a particular subclass of such orderings for which certain superposition inferences *from* variables in skeletons become impossible.

Given a complete reduction ordering $\succ$ on terms, one may extend $\succ$ to (marked) ground literals L by associating a complexity measure c as follows: $c(L) = (\max(L), P, V, \min(L))$, where (i) P is 1 if the literal is negative, and 0, otherwise; and (ii) V is 1 if the root position of the maximal term $\max(L)$ of L belongs to the substitution part of L, and V is 0, otherwise. With this, we define $L \succ L'$ whenever $c(L) \succ c(L')$, where the quadruples are compared lexicographically, using $\succ$ for terms and $1 > 0$ for the bits P and V. This gives us only a partial ordering on marked literals (it is possible for two literals that differ in their markings to have the same complexity measure associated) which we assume to be extended to a total ordering in an arbitrary, but well-founded, manner. The V bit makes literals where the maximal term lies entirely within the substitution part larger than other literals with the same maximal term and polarity.

Let us illustrate this class of literal orderings in terms of what it implies for order reducibility. If $a \succ b \succ c$, the clause $(f(x) \approx b \ \vee \ f(x) \approx a) \cdot [a/x]$ is reduced with respect to the system $\{fa \approx a\}$, but $z \not\approx c \cdot [fa/z]$ and $(x \approx b \ \vee \ fy \approx a) \cdot [fa/x, a/y]$ are not. In the latter case we observe that $(x \approx b) \cdot [fa/x] \succ (fa \approx a)$ since the redex is an instance of a variable.

The significance of this particular literal ordering lies in the avoidance of positive top-level superposition inferences *from* variables.

THEOREM 7.4 With the specific class of orderings, in $\mathsf{SBS}^{\succ}$ no positive superposition inference from a variable into the topmost position of another positive equation is possible.

Proof. For positive superposition inferences of the form

$$\frac{(C \vee s \approx t) \cdot \sigma \quad (D \vee s' \approx v) \cdot \sigma}{(t \approx v \vee C \vee D) \cdot \sigma}$$

(with $s\sigma = s'\sigma$ and s' not a variable) we observe that s cannot be a variable as otherwise the ordering constraints $(s' \approx v) \cdot \sigma \succ (s \approx t) \cdot \sigma$ for the inference would be violated. In fact, the maximal terms and the polarities of the two literals are identical, while, if s is a variable, the V bit for $(s \approx t) \cdot \sigma$ is 1, but $V = 0$ in $c((s' \approx v) \cdot \sigma)$. □

Superposition through variables is extremely prolific since unification provides for no effective filter in such cases. Superposition into substitution positions, in particular variables, are excluded in SBS, regardless of the ordering. With the specific class of literal orderings (which are based on arbitrary reduction orderings on terms), in addition, no positive superposition inferences *from* a skeleton variable into the topmost position of any other positive equation is required. This observation has been crucial for recent results in (Bachmair et al. 1997) on equality elimination transformations suitable for model elimination provers. In the latter paper we may safely adopt this variant of positive strict basic superposition:

$$\frac{(C \vee s \approx t) \cdot \sigma \quad (D \vee w[s'] \approx v) \cdot \sigma}{(w[t] \approx v \vee C \vee D) \cdot \sigma}$$

> where $s\sigma = s'\sigma$, and (i) $s\sigma \succ t\sigma$, (ii) $w\sigma \succ v\sigma$, (iii) $(s \approx t) \cdot \sigma \succ C \cdot \sigma$, (iv) $(w \approx v) \cdot \sigma \succ D \cdot \sigma$, and $(w\sigma, v\sigma) \succeq_{\mathsf{lex}} (s\sigma, t\sigma)$, and (v) s' is not a variable, and either s is not a variable or else s' is a proper subterm of w,

which is the specialization of the general inference with respect to the specific class of literal orderings.

8 Chaining Calculi

Equality is a special case of a transitive relation. Calculi of ordered chaining have been developed by Bachmair & Ganzinger (1994*c*) which may be viewed as generalizations of superposition to possibly non-symmetric transitive relations, including partial and total orderings. Many of these calculi contain an inference, called transitivity resolution, which generalize equality factoring to the non-symmetric case. With the concept of weak counterexamples, one may get rid of transitivity resolution just like we were able to eliminate equality factoring from superposition. For general transitive relations R, chaining from and/or into variables cannot be avoided. Hence, for the general case that we treat below, there is no technical problem related to lifting so that the markings in clause can be ignored, and a setup with closures is not required when presenting the inference systems. However, in the presence of additional axioms for R, some, if not all, chaining inferences through variables are redundant. For those optimized calculi, one again needs to resort to specific orderings on literals that depend on the distinction between substitution and skeleton positions, cf. (Bachmair & Ganzinger 1995) for details.

In the Figure 4 we present (the ground version of) a calculus of ordered chaining OC for one transitive relation S. In a chaining or resolution inference, the first premise is called the positive premise, and the second premise is called the negative premise of the inference. We now assume that we have an arbitrary signature of predicate symbols, including the binary symbol S which is assumed to be a transitive relation. Non-S atoms are eliminated by ordered resolution. OC is parameterized by a well-founded, total ordering on ground terms. The concept of admissible orderings is now slightly different from the equational case. A well-founded ordering $\succ$ on ground literals and ground terms is called admissible if (i) $\succ$ is a total ordering on ground literals, (ii) $\succ$ is a total ordering[3] on ground terms, and (iii) if L and L' are two S literals, then $L \succ L'$, whenever (iii.1) $\max(L) \succ \max(L')$, or (iii.2) $\max(L) = \max(L')$, L is a negative, and L' a positive literal, or (iii.3) $\max(L) = \max(L')$, L and L' have the same polarity, and $\max(L)$ occurs as the left argument of S in L but does not occur as the left argument of S in L'. (Again, by $\max(L)$ we denotes the maximal term (in $\succ$) of an S atom in a literal L). Since S is generally non-symmetric, admissible literal orderings also consider the argument position of the maximal term in an S literal. Otherwise there are no requirements about literal orderings. In particular, non-S atoms may be compared arbitrarily. As before, an admissible clause ordering is the multi-set extension of an admissible literal ordering, and we will only consider admissible orderings on literals and clauses.

We will show that the calculus is reductive for weak counterexamples. For this purpose we provide a model functor similar to what we did in the equational case. The construction requires an adaptation of the notion of a rewrite proof. Let I be a set of ground atoms over the given signature. An atom A has a rewrite proof in I if either (1) A is in I; or else (2) $A = S(s,t)$ and there exist ground terms $s = s_0, \ldots, s_k = t_n, \ldots, t_0 = t$, such that (i) $s_i \succ s_{i+1}$, (ii) $t_{j+1} \prec t_j$, (iii) the rules $S(s_i, s_{i+1})$ and $S(t_{j+1}, t_j)$ which are used in the proof are contained in I. We shall write $s \Downarrow_I t$, whenever $S(s,t)$ has a rewrite proof in I. An atom is called true in I if there exists a rewrite proof for it in I, and is called false in I, otherwise. The extension of this notion of truth to literals and clauses is as usual. A clause which is false in I is also called a counterexample for I. A clause C is called a weak counterexample for I if for any negative literal $\neg A$ in C, A is true in I, and if for no positive literal B in C there exists a direct rewrite proof in I. As in the equational case, in a direct rewrite proof all rules that are used in the proof have to be smaller or equal to the atom being proved.

In short, given the notion of rewrite proof for general transitive relations, the concept of weak counterexample is otherwise the same as for the equational case: for verifying positive literals, one may not rewrite with an R atom that is larger that the fact to be proved.

The definition of the model functor I^w is now essentially the same as for equational clauses and weak counterexamples. Only the concept of a productive clause has to be adapted appropriately.

[3] In the absence of subterm chaining inferences, $\succ$ need not necessarily be a reduction ordering on terms.

Let J be an interpretation. A clause C of the form $D \vee s \approx t$ is called productive on J if
(i) $A \succ C$
(ii) D is a weak counterexample for J, and
(iii) if $A = S(s,t)$, with $s \succ t$, then $u \Downarrow_J t$, for any u such that $s \succ u$ and $S(u,s)$ is in J.
We also say that C produces A from J. With this, we inductively define I_N as

$$I_N = \{A \mid \exists C \in N : C \text{ produces } A \text{ from } I_{N_C}\}.$$

Again, whenever the set N is known from the context or assumed to be arbitrary then we will also omit the index N and also simply write I^w_C for $I^w_{N_C}$. Note that the set of atoms $S(s,t)$ which have a rewrite proof in I^w_N represents a transitive relation. S atoms that would result in a non-convergent "peak" $S(u,s); S(s,t)$ are, due to condition (iii), not produced. Note that atoms of the form $S(u,s)$, with $s \succ u$, do not require an analogue of (iii) since no atom $S(s,t)$, with $s \succ t$, can be smaller than $S(u,s)$ in any admissible literal ordering. The technicalities of the clause ordering imply that if D is a clause in N, and if C and D' are other clauses such that $D' \succ D \succ C$, then C is a weak counterexample for I^w_D if, and only if, C is a weak counterexample for $I^w_{D'}$.

Theorem 8.1 Let N be a set of ground clauses not containing the empty clause. Let C be the minimal weak counterexample in N for I^w_N. Then there exists an inference in OC from C such that
(i) its conclusion is a weak counterexample for I^w_N and is smaller than C; and
(ii) if the inference is a chaining inference or a resolution inference then C is its negative premise and the positive premise is a productive clause.

This theorem can be again be easily verified by an inspection of all the possible forms of counterexamples.

The theorem in part (i) states that S has the reduction property, and hence is refutationally complete and compatible with the removal of redundant clauses if redundancy is defined by the criterion $\mathcal{R}^w$. In that definition the concept of a convergent ground rewrite system has to be replaced by Herbrand interpretations R in which the set of S atoms that have a rewrite proof in R is a transitive relation. Part (ii) of the theorem gives further semantic restrictions to OC which may be exploited with a notion of redundancy for inferences in addition to the one for clauses.

Theorem 8.2 N be saturated up to redundancy with respect to OC. Then N has a model in which S is a transitive relation if, and only if, N does not contain the empty clause.

Positive chaining

$$\frac{C \vee S(t,s) \quad D \vee S(s,v)}{S(t,v) \vee C \vee D}$$

where (i) $s \succ t$, (ii) $s \succ v$, (iii) $S(t,s) \succ C$, and (iv) $S(s,v) \succ D$.

Negative chaining

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

where (i) $s \succ t$, (ii) $s \succeq v$, (iii) $S(s,t) \succ C$, and (iv) $\neg S(s,v) \succeq \max(D)$;

and

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

where (i) $s \succ t$, (ii) $s \succeq v$, (iii) $S(t,s) \succ C$, and (iv) $\neg S(v,s) \succeq \max(D)$.

Ordered resolution

$$\frac{C \vee A \quad D \vee \neg A}{C \vee D}$$

where (i) $A \succ C$, and (ii) $\neg A \succeq \max(D)$.

Ordered Factoring

$$\frac{C \vee A \vee A}{C \vee A}$$

where $A \succeq \max(C)$.

Figure 4: Ordered Chaining OC

9 Conclusions

This paper solves a long-standing open problem showing that the superposition calculus by Zhang & Kapur (1988) is refutationally complete if certain tautologies are not eliminated. More precisely, we have shown that slight restrictions of the usual simplification and redundancy elimination techniques are compatible with strict basic superposition, a substantially more restrictive form of ZK. The completeness results have been proved for large classes of admissible orderings on terms and literals. In particular one may choose literal orderings which generally avoid positive superposition inferences *from* variables into the top position of positive equations. The results have also been extended to ordered chaining, showing that transitivity resolution and composition resolution inferences are not required.

On the methodological level we have been able to make our previous proof techniques more transparent. The two main components in such proofs, reduction of counterexamples through inferences and the identification of clauses that cannot become counterexamples have been strictly separated. Constructions similar to our candidate models for superposition have been sketched, but not formalized with the required mathematical rigor, by Zhang (1988). Already Brand's proof in (1975) of his equality elimination method contains related ideas. The present definitions, including redundancy, have originated from (Bachmair & Ganzinger 1990), where, however, proofs are technically more difficult in that they deal with counterexample reduction and redundancy simultaneously. Shortly after, Pais & Peterson (1991) have given a similar proof of completeness of superposition, however, without presenting a general concept of redundancy. For SBS the semantic concept of counterexamples as suitable for non-strict superposition needed to be replaced by a proof-theoretic notion based on direct rewrite proofs.

From a theoretical point of view, strict basic superposition is rather appealing in that the two main issues in equational clause logic, transitivity and disjunction, are now clearly separated. As to whether strict basic superposition is the method of choice in practice still needs to be seen. Due to the technical complications with the basic setup and with simplification, implementing the calculus efficiently appears to be a non-trivial task.

References

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

Bachmair, L. & Ganzinger, H. (1994*a*), Ordered chaining for total orderings, *in* 'Proc. 12*th* International Conference on Automated Deduction', Vol. 814 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, pp. 435–450.

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

Bachmair, L. & Ganzinger, H. (1994*c*), Rewrite techniques for transitive relations, *in* 'Proc. 9th IEEE Symposium on Logic in Computer Science', IEEE Computer Society Press, pp. 384–393.

Bachmair, L. & Ganzinger, H. (1995), Ordered chaining calculi for first-order theories of binary relations, Technical Report MPI-I-95-2-009, Max-Planck-Institut für Informatik, Saarbrücken. Revised version to appear in JACM.

Bachmair, L., Ganzinger, H., Lynch, C. & Snyder, W. (1992), Basic paramodulation and superposition, *in* D. Kapur, ed., 'Automated Deduction — CADE'11', Vol. 607 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 462–476.

Bachmair, L., Ganzinger, H. & Voronkov, A. (1997), Equality elimination with ordering constraints, Technical Report MPI-I-97-2-012, Max-Planck-Institut für Informatik, Saarbrücken.

Brand, D. (1975), 'Proving theorems with the modification method', *SIAM J. Comput.* **4**, 412–430.

Nieuwenhuis, R. & Rubio, A. (1992*a*), Basic superposition is complete, *in* 'ESOP'92', Vol. 582 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 371–389.

Nieuwenhuis, R. & Rubio, A. (1992*b*), Theorem proving with ordering constrained clauses, *in* 'Automated Deduction — CADE'11', Vol. 607 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 477–491.

Pais, J. & Peterson, G. (1991), 'Using forcing to prove completeness of resolution and paramodulation', *Journal of Symbolic Computation* **11**, 3–19.

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

Zhang, H. & Kapur, D. (1988), First-order theorem proving using conditional rewrite rules, *in* E. Lusk & R. Overbeek, eds, 'Proc. 9th Int. Conf. on Automated Deduction', Vol. 310 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 1–20.

I N F O R M A T I K

Below you find a list of the most recent technical reports of the research group *Logic of Programming* at the Max-Planck-Institut für Informatik. They are available by anonymous ftp from our ftp server `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Birgit Hofmann
Im Stadtwald
D-66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-97-2-010	S. Vorobyov, A. Voronkov	Complexity of Nonrecursive Logic Programs with Complex Values
MPI-I-97-2-009	A. Bockmayr, F. Eisenbrand	On the Chvátal Rank of Polytopes in the 0/1 Cube
MPI-I-97-2-008	A. Bockmayr, T. Kasper	A Unifying Framework for Integer and Finite Domain Constraint Programming
MPI-I-97-2-007	P. Blackburn, M. Tzakova	Two Hybrid Logics
MPI-I-97-2-006	S. Vorobyov	Third-order matching in $\lambda \rightarrow$-Curry is undecidable
MPI-I-97-2-005	L. Bachmair, H. Ganzinger	A Theory of Resolution
MPI-I-97-2-004	W. Charatonik, A. Podelski	Solving set constraints for greatest models
MPI-I-97-2-003	U. Hustadt, R.A. Schmidt	On evaluating decision procedures for modal logic
MPI-I-97-2-002	R.A. Schmidt	Resolution is a decision procedure for many propositional modal logics
MPI-I-97-2-001	D.A. Basin, S. Matthews, L. Viganò	Labelled modal logics: quantifiers
MPI-I-96-2-010	A. Nonnengart	Strong Skolemization
MPI-I-96-2-009	D.A. Basin, N. Klarlund	Beyond the Finite in Automatic Hardware Verification
MPI-I-96-2-008	S. Vorobyov	On the decision complexity of the bounded theories of trees
MPI-I-96-2-007	A. Herzig	SCAN and Systems of Conditional Logic
MPI-I-96-2-006	D.A. Basin, S. Matthews, L. Viganò	Natural Deduction for Non-Classical Logics