Strict Basic Superposition

Leo Bachmair*[1] and Harald Ganzinger**[2]

[1] Comp. Sci. Dept., SUNY at Stony Brook, NY 11794, U.S.A., leo@cs.sunysb.edu
[2] MPI Informatik,D-66123 Saarbrücken, Germany, hg@mpi-sb.mpg.de

Abstract. 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 in terms of, not semantic truth, but direct provability. We introduce a corresponding concept of redundancy with which strict superposition is compatible and that covers most simplification techniques. We also show that certain superposition inferences *from* variables are redundant—a result that is relevant, surprisingly, in the context of equality elimination methods.

1 Introduction

Superposition, the main deductive inference rule in completion, combines unification with the replacement of subterms by equal terms. In ordered completion, which is the variant relevant for theorem proving, the key restriction imposed on inferences is that only *maximal* sides of equation be used to replace subterms of *maximal* sides of other equations, where terms are compared in a given ordering. Completion deals only with unit equations, but various extensions of superposition to non-unit clauses have been proposed, see (Bachmair & Ganzinger 1994) for a detailed discussion. The natural additional restrictions on clausal inferences are that the literals involved in an equational replacement be *maximal* in their respective clauses. But proving that such a restricted version of paramodulation (which superposition is in the clause context) is refutationally complete turned out to be an elusive problem.

The superposition calculi introduced in Bachmair & Ganzinger (1990) impose all the restrictions, but contain additional inference rules, such as equality factoring, that need to be applied to certain (non-Horn) clauses. The calculus ZK proposed by Zhang & Kapur is closely related to superposition. It is less

* Work supported in part by NSF under grants INT-9314412 and CCR-9510072.
** Work supported in part by DFG under grants Ga 261/7-1 and Ga 261/8-1, and by the ESPRIT Basic Research Working Group 22457 (CCL II).

restrictive in that replacements are allowed in non-maximal terms of maximal negative equations, but on the other hand contains no equality factoring or similar inference rule. 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 tautology deletion renders ZK incomplete. Since then refutational completeness of ZK when tautologies are not eliminated, has remained an open problem.

In (Bachmair & Ganzinger 1990, Bachmair & Ganzinger 1994) it was shown that superposition can be combined with a variety of simplification techniques including those known from completion. A general notion of redundancy based on logical entailment from smaller clauses justifies the admissibility of these techniques. In this context tautologies are always redundant and can be deleted at any point during the theorem proving process. The proof techniques developed in these papers are therefore not directly applicable to ZK. It was also not clear to what extent other simplification techniques, such as subsumption or (standard or contextual) rewriting, are compatible with ZK. (The question is especially important from a practical point of view, as calculi that do not allow reasonably strong simplification mechanisms are virtually useless for automated theorem proving.)

In this paper we study strict superposition – superposition without equality factoring. We prove that this calculus, which is a substantially more restrictive version of ZK, is not only refutationally complete, but is also compatible with a notion of redundancy that covers most, though not all, of the major simplification techniques. Our results are obtained by introducing a suitable notion of *direct rewrite proof* and modifying proof techniques based on candidate models and counterexamples. 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 the two central concepts: (i) the reduction of counterexamples by inferences, and (ii) notions of redundancy that provide approximate characterizations of clauses that cannot be minimal counterexamples. Reasoning about strict superposition requires "weak" counterexamples and, therefore, a weaker notion of redundancy. Lifting is more complicated for strict superposition and requires the techniques of basic superposition (Bachmair, Ganzinger, Lynch & Snyder 1992, Nieuwenhuis & Rubio 1992*a*). We describe the basic ideas and concepts; proofs can be found in (Bachmair & Ganzinger 1997).

2 Preliminaries

We assume the usual notions and notations about equational clauses, rewrite systems, and reduction orderings. For simplicity, we assume that equality, denoted by the symbol $\approx$, is the only predicate. (Equations are taken to be symmetric, so that we do not distinguish between atoms $s \approx t$ and $t \approx s$.) An *equality (Herbrand) interpretation* is a congruence on ground terms over the given signature.

Let $\succ$ be a reduction ordering (on terms). We say that a set of ground equations I is a *convergent ground rewrite system* (with respect to $\succ$) if (i) all equations $s \approx t$ in I are orientable with respect to $\succ$ (i.e., either $s \succ t$ or $t \succ s$) and (ii) the corresponding set of oriented *rewrite rules* is confluent (in the usual sense). We use such convergent ground rewrite systems to represent equality interpretations. An atom $s \approx t$ is true in the *interpretation represented by I* if, and only if, there exists a *rewrite proof* for $s \approx t$ (written $s \Downarrow_I t$). For simplicity, we will usually speak of the "interpretation I" when in fact we mean the interpretation *represented by I*.

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.[1] If $\succ$ is an admissible ordering, its multi-set extension (which we usually denote by the same symbol) is a well-founded ordering on clauses that is total on ground clauses. We call such clause orderings *admissible* as well. 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. By condition (iii.2) clauses in which the maximal term occurs in a negative literal are larger than clause with the same maximal term, but occurring only in positive literals. Condition (iii) circumscribes the relationship between the term and literal orderings, but leaves some flexibility for extending a given term ordering to literals. For instance, comparisons between literals with the same maximal may take into account the respective minimal terms.

The calculi in this paper are parameterized by an admissible clause ordering. From now on, let $\succ$ denote an arbitrary, but fixed admissible ordering.

3 Strict and Non-Strict Superposition

We first consider ground clauses only. The extension of our results to general clauses is discussed in Section 7.

The inference rules of *strict (ground) superposition* SS are shown in Figure 1.[2] The ordering constraints imposed on superposition inferences restrict equational replacements so that only maximal terms of maximal equations in the second premise (the *negative premise*) are replaced by the minimal term of the maximal equation of the first (the *positive*) premise.

The calculus ZK (Zhang & Kapur 1988) is a predecessor of SS. The positive superposition rule in ZK imposes constraints (i) and (ii), but not (v), whereas (iii) and (iv) are slightly different due to differences in the literal orderings. For negative superposition the important restriction (ii) is dropped, and (unordered) factoring may be applied to both positive and negative literals.

[1] Here $\max(L)$ denotes the maximal term (with respect to $\succ$) of a literal L. We will also use $\max(D)$ to denote the maximal literal in a clause D.

[2] In order to simplify matters technically, we do not consider selection functions on negative literals. But all the results can be extended appropriately.

Positive (ground) 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 (ground) 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 $(s \approx t) \succeq \max(C)$.

Fig. 1. Strict (ground) superposition calculus SS

Strict superposition is not compatible with tautology elimination. For example, the set of clauses

$$\begin{array}{r} a \not\approx b \vee a \approx c \\ b \approx c \\ a \not\approx b \vee a \not\approx d \end{array}$$

where a, b, and c are constants, is unsatisfiable. However, if we choose an ordering $a \succ b \succ c$, then only the tautology $b \not\approx b \vee a \not\approx c \vee a \approx c$ can be derived by strict superposition.[3] In other words, strict superposition is not refutationally complete in combination with tautology deletion. In the less restrictive calculus ZK the empty clause can be derived from these clauses, even when one deletes tautologies. But the following example from (Bachmair & Ganzinger 1990) demonstrates that ZK is not compatible with tautology deletion either:

$$\begin{array}{ll} c \not\approx d & \\ & b \approx d \\ a \not\approx d \vee a \approx c & \\ & a \approx b \vee a \approx d \end{array}$$

[3] This example was communicated to us by Chris Lynch.

If $a \succ b \succ c \succ d$, only the tautology $a \not\approx d \vee b \approx c \vee a \approx d$ can be obtained by ZK.[4]

In (Bachmair & Ganzinger 1990) we had introduced extended superposition calculi that are refutationally complete and compatible with tautology deletion. The *(non-strict) superposition* calculus S consists of SS plus the following *equality factoring* rule:

Equality factoring

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

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

Equality factoring combines resolution (of a clause with a certain instance of the transitivity axiom) and factoring as the following derivation shows:

$$\frac{\dfrac{C \vee s \approx t \vee s \approx t' \quad (x \not\approx y \vee y \not\approx z \vee x \approx z)[t'/z]}{C \vee t \not\approx t' \vee s \approx t' \vee s \approx t'}}{C \vee t \not\approx t' \vee s \approx t'}$$

The inference rule obviously applies only to non-Horn clauses.

4 Candidate Models and Counterexamples

A powerful method for proving the refutational completeness of deductive calculi is centered around the concept of candidate models and reduction of counterexamples. The main feature in our current presentation is the clear separation between *local* aspects of reducing counterexamples (by inferences) and *global* methods for characterizing counterexamples. The latter also lead to general notions of redundancy: a clause is inherently redundant if it can never become a minimal counterexample.

4.1 Basic Concepts

If $\mathcal{I}$ is an (clausal) inference system, we denote by $\mathcal{I}(N)$ the set of all clauses that can be derived by applying an inference in $\mathcal{I}$ to premises in N. Suppose we have a mapping I, called a *model functor*, that assigns to each set N of ground clauses not containing the empty clause, an (equality) Herbrand interpretation I_N, called a *candidate model* (for N). If I_N is indeed a model of N, then N is satisfiable. Otherwise, N contains clauses, called *counterexamples* (for I_N), that are false in I_N. If a counterexample exists at all, then N also contains a minimal counterexample with respect to the clause ordering $\succ$. We say that $\mathcal{I}$ has the *reduction property for counterexamples* (with respect to I) if, whenever C is a

[4] ZK employs a slightly different literal ordering in which the polarity of a literal is ignored, so that $a \approx c \succ a \not\approx d$. These technical differences are not significant, though.

minimal counterexample for I_N, then there exists an inference in $\mathcal{I}$ from C (and possibly additional premises from N) with a conclusion D that is smaller than C (i.e., $C \succ D$) and is also a counterexample for I_N. Inference systems that have the reduction property for counterexamples are refutationally complete in the following sense.

Proposition 1. *If $\mathcal{I}$ has the reduction property for counterexamples and if N is closed under $\mathcal{I}$, that is, $\mathcal{I}(N) \subseteq N$, then N either contains the empty clause, or else is satisfiable.*

Our approach to proving refutational completeness is based on establishing the reduction property for counterexamples. A key advantage of this approach is that it naturally leads to a very general notion of redundancy.

First observe that only inferences that reduce minimal counterexamples are essential. A clauses that can never become a minimal counterexample is inherently redundant. Of course, the difficulty is to characterize such clauses. The following approximate characterization captures many, but not all, inherently redundant clauses.

We say that a ground clause C (not necessarily in N) is *redundant* with respect to a set N of ground clauses if there exist clauses $C_1, \ldots, C_k$ in N such that $C_1, \ldots, C_k \models C$ 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 $\mathcal{I}$ if $\mathcal{I}(N \setminus \mathcal{R}(N)) \subseteq N \cup \mathcal{R}(N)$. Let us also denote by N_C the set of clauses D in N with $C \succ D$. 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. Note that the admissible ordering $\succ$ provides the link between the redundancy criterion $\mathcal{R}$ and the inference system $\mathcal{I}$.

Proposition 2. *Let $\mathcal{I}$ be a sound inference system with the reduction property. If N is saturated up to redundancy with respect to $\mathcal{I}$, then it is unsatisfiable if, and only if, it contains the empty clause.*

In previous papers, e.g., (Bachmair & Ganzinger 1994), we also formalized the notion of theorem proving processes, as sequences of individual steps of deduction (of new clauses) and deletion (of redundant clauses). Under certain fairness assumptions, theorem proving sequences result in saturated clause sets, and thus represent refutationally complete theorem proving strategies. (A key property is that redundancy of clauses is preserved under deduction and deletion, so that redundant clauses can always be eliminated.) These investigations are essential for effectively and efficiently constructing saturated sets of clauses, but orthogonal to the topic of the present paper. The key question we consider next is how to construct a model functor I so that a given inference system $\mathcal{I}$ has the reduction property for counterexamples with respect to I.

4.2 Constructing Candidate Models

The definition of the model functor I will be monotone for downward closed subsets of N. More precisely, whenever C and D are two clauses with $C \preceq D$, then $I_{N_C} \subseteq I_{N_D} \subseteq I_N$. The sets I_{N_C} will be defined by induction over the given clause ordering, via auxiliary sets of rewrite rules $E_{N,C}$ designed so that in many cases a clause C true in $I_{N_C} \cup E_{N,C}$.

Let C be any given ground clause (not necessarily in N) and suppose that $E_{N,C'}$ and $I_{N_{C'}}$ have already been defined for all clauses C' for which $C \succ C'$. Then

$$I_{N_C} = \bigcup_{C \succ C'} E_{N,C'}.$$

Moreover, $E_{N,C} = \{s \approx t\}$ if C is a clause in N of the form $D \vee s \approx t$ such that (i) D is a counterexample for $I_{N_C} \cup \{s \approx t\}$, (ii) $(s \approx t) \succ D$ and $s \succ t$, and (iii) s is irreducible by I_{N_C}. In that case, we also say that C *produces* the equation (or rule) $s \approx t$. In all other cases, $E_{N,C} = \emptyset$. Finally, we define $I_N = \bigcup_C E_{N,C}$ as the set of all equations produced by the clauses in N.[5] Whenever the set N is known from the context or assumed to be arbitrary then we will omit the index N from I_N, I_{N_C}, and $E_{N,C}$.

Clauses that produce equations are called *productive.* We say that a clause $C \vee s \approx t$ is *reductive for* $s \approx t$ if $(s \approx t) \succ C$ and $s \succ t$. Productive clauses are reductive. Only clauses in N can produce an equation for I_N. A productive clause C is false in I_C, but true in $I_C \cup E_C$, whereas the subclause D remains a counterexample for $I_C \cup E_C$. By condition (ii) the maximal term in a productive clause cannot appear in a negative literal. The sets I_C and I_N are constructed in such a way that they are left-reduced rewrite systems with respect to $\succ$ and, therefore, convergent. It can also be shown 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 true in I_D if, and only if, it is true in $I_{D'}$.

Let us explain this definition 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 descending order from the top) plus the rewrite rules that are produced.

C	E_C	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
$\rightarrow c \approx d$	$c \approx d$	reductive and false in I_C

All but the second clause are true in the candidate model $I = \{c \approx d,\ a \approx c\}$. A negative superposition inference

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

between the third (productive) clause and the counterexample yields a smaller counterexample $c \approx d,\ c \approx d \rightarrow b \approx c,\ b \approx c$ for I. In general, if a clause C such as

[5] Note that I is well-defined; that is, if $N = M$ then $I_N = I_M$. This can be easily verified by induction over the clause ordering.

$a \approx d \to b \approx c$ is false in I_C, its maximal term a must be reducible by a rule in I_C (that was produced by some clause C' in N_C). Thus there is a negative superposition inference with a conclusion smaller than C. Moreover, condition (i) ensures that this conclusion is also a counterexample. Other kinds of counterexamples can be reduced in a similar way (but by different inferences).

Theorem 1. *Let N be a set of ground clauses not containing the empty clause and 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 that 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.*

Part (i) of the theorem states that S has the reduction property and, hence, is refutationally complete and compatible with the removal of redundant clauses. Part (ii) provides further restrictions that may be used to define a notion of redundancy for inferences.

Let us now apply the construction to the critical example from Section 3.

clauses C	E_C	remarks
$a \approx b,\ a \approx c \to$		true
$\to a \approx b,\ a \approx c$		minimal counterexample
$\to b \approx c$	$b \approx c$	reductive and false in I_C

The clause $D = a \approx b,\ a \approx c$ is non-productive, for if $a \approx b$ were produced, then $a \approx c$ would become true and condition (i) be violated. Equality factoring can be used to reduce this kind of counterexample. We have seen in Section 3 that a contradiction can also be derived if one computes inferences with tautologies. But this alternative requires that we relax the notion of counterexamples so that certain tautologies can be considered "weak" counterexamples for some interpretations. The modified, proof-theoretic techniques 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.

For example, if R contains two equations, $a \approx b$ and $b \approx c$, where $a \succ b \succ c$, then $a \approx c$ has a rewrite proof in R, but not a direct rewrite proof. In a direct rewrite proof no rewrite rules may be used that are 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.

We say an inference system $\mathcal{I}$ has the *reduction property for weak counterexamples* (with respect to a model functor 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) with a conclusion D that is smaller than C and also a weak counterexample for I_N.

Proposition 3. *If $\mathcal{I}$ has the reduction property for weak counterexamples and N is closed under $\mathcal{I}$, that is, $\mathcal{I}(N) \subseteq N$, then N either contains the empty clause or else is satisfiable.*

Weak counterexamples suggest an obvious modification in the notion of redundancy. A ground clause C (not necessarily in N) is said to be *(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 no clause in $N \setminus \mathcal{R}^w(N)$ is a weak counterexample for an interpretation R then no clause in N can be a weak counterexample for R either, and, hence, 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 4. *Let $\mathcal{I}$ be a sound inference system with the reduction property for weak counterexamples. If a set N is saturated up to weak redundancy with respect to $\mathcal{I}$, then it is unsatisfiable if and only if it contains the empty clause.*

In applying these concepts to strict superposition, we modify the previous definition of a model functor I_N by replacing "counterexample" by "weak counterexample". Let I_N^w denote the modified functor. Note that the modified condition (i) for productive clauses C essentially requires that C be a weak counterexample for I_C^w. (If $C = D \vee s \approx t$ is a weak counterexample for I_C^w, no positive literal in D has a direct rewrite proof in $I_C^w \cup \{s \approx t\}$, even if it is of the form $s \approx u$. As the clause is reductive for $s \approx t$, we have $(s \approx t) \succ (s \approx u)$ so that $s \approx t$ cannot be applied in any direct rewrite proof of $s \approx u$.)

We now obtain a different candidate model $I^w = \{a \approx b,\ b \approx c\}$ for the above example

C	I^w	remarks
$a \approx b,\ a \approx c \rightarrow$		minimal weak counterexample
$\rightarrow a \approx b,\ a \approx c$	$a \approx b$	reductive and false in I_C
$\rightarrow b \approx c$	$b \approx c$	reductive and false in I_C

The clause $a \approx b, \vee\ a \approx c$ is productive now and may be used to reduce the counterexample to a tautology $b \approx b,\ a \approx c \rightarrow a \approx c$ that is a smaller weak counterexample for I^w! In fact, strict superposition does have the reduction property for weak counterexamples and, hence, is refutationally complete for ground clauses.

Theorem 2. *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 *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.*

6 Simplification Afforded by Weak Redundancy

We next briefly investigate to what extent the main simplification techniques in superposition theorem proving can be used in the context of weak redundancy. We discuss the ground case. For the non-ground case additional consideration have to be taken into account, cf. Section 7.

In general, simplification is a derivation step (on sets of clauses) of the form $N \cup \{C\} \vdash N \cup \{D\}$ where C and D are logically equivalent clauses with $C \succ D$, and C is in $\mathcal{R}^w(N \cup \{D\})$. In other words, simplification allows one to replace a clause C by a simpler clause D, provided C is redundant.

Simplification by standard term rewriting is admissible.

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 (i) $s \succ t$, (ii) $w[s] \approx u \succeq s \approx t$, and (iii) $w[s] \approx u \succ w[t] \approx u$.

Soundness is trivial. The clause $C \vee w[s] \approx u$ is redundant in $\{s \approx t,\ C \vee w[t] \approx u\}$ as direct rewrite proofs of $s \approx t$ and $w[t] \approx u$ with respect to a convergent rewrite system can be combined into a direct rewrite proof of $w[s] \approx u$. Note that, depending on the literal ordering, the constraint (iii) is not necessarily implied by (i) and (ii). Simplifying a negative literal is essentially similar, except that the constraint (ii) is redundant.

Tautology elimination necessarily has to be weaker than usual. Here is an admissible variant.

Direct tautology elimination:

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

if C is of the form $C_n \vee C_p \vee w[s] \approx t$, where (i) C_n the subclause of all negative literals in C, (ii) $w[s] \neq s$ and $w[s] \succeq t$, and (iii) there exists a term u with $s \succ u$, such that both $C_n \vee s \approx u$ and $C_n \vee w[u] \approx t$ are tautologies.

With the specified restrictions, any convergent rewrite system in which the equations in C_n are true admits a rewrite proof of $w[s] \approx w[u]$ (hence of $w[s] \approx t$) using only rules in which the maximal term is smaller than $w[s]$.

Proper subsumption works as usual.

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

In sum, we have shown that with certain additional restrictions, the usual simplification techniques for standard superposition are also admissible for the strict variant.

7 Lifting

In refutational theorem proving, one usually reasons about non-ground clauses by considering their ground instances. As shown in the full paper (Bachmair & Ganzinger 1997), equational replacements in the "substitution part" of a weak counterexample might result in clauses that are no longer weak counterexamples. Therefore, the lifting argument for standard superposition does not work, and we need to restrict ourselves to "reduced" instances of clauses, like in basic superposition (Bachmair et al. 1992, Nieuwenhuis & Rubio 1992*a*).

Basic strategies are conveniently formulated with constrained clauses (Nieuwenhuis & Rubio 1992*b*), but since in this paper we are mainly concerned with a solution to the lifting problem, we shall only employ a simple marking schema to specify those subterm positions in ground clauses on which equational replacements are not allowed. We essentially need to distinguish between (unmarked) positions that were present in the original clauses, and (marked) positions that were introduced by substitutions applied as part of inferences. For that purpose, we use the notation $C \cdot \sigma$, where C is a non-ground clause and σ a ground substitution, to denote a ground instance $C\sigma$. Expressions $C \cdot \sigma$ are also called *closures*. The clause part C of a closure is also called its *skeleton*.[6] Positions in $C\sigma$ that are at or below a variable in C are called *substitution positions* (or *marked* positions). Admissible orderings are defined as before, though an ordering may now depend on the distinction between unmarked and substitution positions (in cases where the maximal terms and the polarity of two literals are the same, cf. Section 8).

A notion central to the lifting of basic inference systems is that of a reduced closure. We say that a literal L is *order-reducible* (at position p) by an equation $s \approx t$, if s is the subterm of L at position p, $s \succ t$, 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 $s \approx t$. A positive literal $u \approx v$ is order-reducible by $s \approx t$ only if $(u \approx v) \succ (s \approx t)$ in the literal ordering.) A literal is *order-reducible by R* if it is order-reducible by some equation in R. A ground clause is said to be *reduced* with respect to R if it is not order-reducible with respect to R at any substitution position.

The basic variant of positive superposition can be formulated as follows:

Positive 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 $(w \approx v) \cdot \sigma \succ (s \approx t) \cdot \sigma$, and (v) s' is not a variable

[6] We assume that the variables in the *skeleton* C of a closure $C \cdot \sigma$ are α-convertible to achieve variable disjointness wherever needed. In particular, any two closures can always be written as $C \cdot \sigma$ and $D \cdot \sigma$, respectively, with a common substitution σ.

This is essentially the standard superposition rule applied to closures. Other inference rules are obtained in a similar way from the non-basic rules. Note that condition (v) ensures that subterm replacements never take place at substitution positions. Another characteristic property of the basic variant is that skeleton positions in the conclusion correspond to skeleton positions in the premises so that no new skeleton positions are ever generated.

In investigating the reduction property for SBS we assume that candidate models I^w for sets of ground clauses N are defined as before. Produced equations inherit their marked substitution positions from the producing clause.

A set of ground clauses N is said to be *reduced* if any clause C in N is reduced with respect to I^w_C. By the monotonicity of I^w, and properties of the clause orderings, we observe that if C is reduced with respect to I^w_C it is also reduced with respect to 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* if it contains all clauses in N that are reduced with respect to I^w_M.

Proposition 5. *Any set of clauses contains a maximal reduced subset.*

A maximal reduced subset M is obtained (inductively) as the set of all clauses C such that C is reduced with respect to $I^w_{M_C}$.

Theorem 3. *Let N be a reduced set of clauses not containing the empty clause and 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 that is smaller than C and 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.*

Part (i) states that SBS has the reduction property for weak counterexamples in reduced sets of closures. It in addition asserts that the smaller counterexample is again a reduced closure, which is important for effective saturation.

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

Theorem 4. *Let N be a set of closures that is saturated up to redundancy with respect to* SBS *and contains a schematic subset K, such that every closure in*

$N \setminus K$ is a logical consequence of K. Then N either contains the empty closure, or else is satisfiable.

Whenever N is the closure under SBS (ignoring inferences involving redundant closures) 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.

8 Optimized Variable Chaining

The preceding completeness results hold for all admissible orderings. We shall now define a particular subclass of such orderings that prohibit certain superposition inferences *from* variables.

First we define a "complexity measure" c on marked literals as follows: $c(L) = (\max(L), P, V)$, where (i) P is 1 if the literal is negative, and 0, otherwise; and (ii) V is 1 if L is of the form $[\neg](x \approx v) \cdot \sigma$, with x a variable and $\max(L) = x\sigma$, and V is 0, otherwise. Such triples are compared lexicographically, using the given term ordering in the first component and using $1 > 0$ for the bits P and V in the other components. We may then consider the class of admissible orderings on marked literals which are compatible with this complexity measure in that $L \succ L'$ whenever $c(L) \succ c(L')$. Note that the V bit makes literals in which the maximal term lies entirely within the substitution part larger than other literals with the same maximal term and polarity.

The significance of these particular literal orderings is that they allow us to exclude positive top-level superposition inferences *from* variables.

Theorem 5. *If $\succ$ is a literal ordering as specified above, then no positive superposition inferences from a variable into the topmost position of another positive equation are possible in $\mathsf{SBS}^{\succ}$.*

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 constraint $(w \approx v) \cdot \sigma \succ (s \approx t) \cdot \sigma$ associated with the inference would be violated. (The maximal terms and the polarities of the two literals in question are identical. But if s is a variable, the V bit in $c((s \approx t) \cdot \sigma)$ is 1, whereas in $c((w \approx v) \cdot \sigma)$ it is 0.

Superposition through variables is extremely prolific since unification is no effective filter in such cases. Superpositions into substitution positions, in particular variables, are excluded in SBS, regardless of the ordering. With the specific class of literal orderings (that are based on arbitrary reduction orderings on terms), no positive superposition inferences *from* a skeleton variable into the topmost position of any other positive equation is required either. This observation has been crucial for recent results on equality elimination transformations (Bachmair, Ganzinger & Voronkov 1997).

9 Conclusions

In this paper we solve a long-standing open problem by showing that the superposition calculus ZK by Zhang & Kapur (1988) is refutationally complete provided certain tautologies are not eliminated. More generally, we have shown that slight restrictions of the usual simplification and redundancy elimination techniques are compatible with strict basic superposition, a substantially more restrictive calculus than ZK. We have proved completeness for a large class of admissible orderings on terms and literals. In particular one may choose literal orderings that avoid positive superposition inferences *from* variables into the top position of positive equations.

The results about strict superposition can be also extended to the case of ordered chaining calculi for general transitive relations. One can show (Bachmair & Ganzinger 1997) that transitivity (or composition) resolution inferences, the corresponding generalizations of the equality factoring inference, as they were required in (Bachmair & Ganzinger 1995) can also be dispensed with at the expense of a weaker notion of redundancy.

On the methodological level, we believe that the presentation in the current paper makes the underlying proof techniques more transparent, as it separates the two main components, reduction of counterexamples by inferences and characterizations, via redundancy, of clauses that cannot become counterexamples. Constructions similar to our candidate models for superposition have been sketched, but not formalized with mathematical rigor, by Zhang (1988). Related ideas can already be found in Brand's proof of his equality elimination method (Brand 1975). The present definitions, including redundancy, originated from (Bachmair & Ganzinger 1990), where, however, the exposition is technically considerably more involved, mainly because counterexample reduction and redundancy are dealt with simultaneously. Pais & Peterson (1991) have given a similar proof of the completeness of superposition with merging paramodulation (a variant that was also introduced in (Bachmair & Ganzinger 1990), but without presenting a general concept of redundancy. Strict superposition, as we have explained above, requires that the semantic concept of counterexamples 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 two critical issues for equational theorem proving, how to deal with transitivity and disjunction, respectively, are clearly separated. It remains to be seen whether strict basic superposition is the method of choice in practice. Due to the technical complications with the basic setup and with restricted simplification, implementing the calculus efficiently appears to be a non-trivial task.

Acknowledgements. We are grateful to Chris Lynch for his comments on this paper and, in particular, for pointing out a bug in the preliminary version.

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), 'Rewrite-based equational theorem proving with selection and simplification', *J. Logic and Computation* **4**(3), 217–247. Revised version of Technical Report MPI-I-91-208, 1991.

Bachmair, L. & Ganzinger, H. (1995), Ordered chaining calculi for first-order theories of binary relations, Research Report MPI-I-95-2-009, Max-Planck-Institut für Informatik, Saarbrücken, Saarbrücken. Revised version to appear in JACM.
URL: www.mpi-sb.mpg.de/~hg/pra.html#MPI-I-95-2-009

Bachmair, L. & Ganzinger, H. (1997), Strict basic superposition and chaining, Research Report MPI-I-97-2-011, Max-Planck-Institut für Informatik, Saarbrücken, Saarbrücken.
URL: www.mpi-sb.mpg.de/~hg/pra.html#MPI-I-97-2-011

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), Elimination of equality via transformation with ordering constraints, Research Report MPI-I-97-2-012, Max-Planck-Institut für Informatik, Saarbrücken, Saarbrücken.
URL: www.mpi-sb.mpg.de/~hg/pra.html#MPI-I-97-2-012

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', *J. 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.