In Proc. CADE-13, New Brunswick, NJ, 1996

# Theorem Proving in Cancellative Abelian Monoids (Extended Abstract)*

Harald Ganzinger and Uwe Waldmann

Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany
E-Mail: {hg,uwe}@mpi-sb.mpg.de, Phone: +49 681 9325 {201,227}

**Abstract.** Cancellative abelian monoids encompass abelian groups, but also such ubiquitous structures as the natural numbers or multisets. Both the AC axioms and the cancellation law are difficult for a general purpose theorem prover, as they create many variants of clauses which contain sums. We describe a refined superposition calculus for cancellative abelian monoids which requires neither explicit inferences with the theory clauses nor extended equations or clauses. Strong ordering constraints allow us to restrict to inferences that involve the maximal term of the maximal sum in the maximal literal. Besides, the search space is reduced drastically by variable elimination techniques.

## 1 Introduction

To be useful in applications such as program verification and synthesis, a theorem prover must combine mathematical with meta-mathematical reasoning. Theories from which program properties are to be derived are divided into parts that specify standard mathematical structures, including numbers, lists, multisets, graphs, and into other parts that provide the axioms for additional free function and predicate symbols. The latter describe, in a more or less ad hoc manner, objects, and their properties, of the particular domain of application for which a program is to be written.

There are several distinct lines of investigation along which previous attempts have been made for attacking this problem. In hierarchic situations where free function symbols either do not exist or where they are defined in a "sufficiently complete" manner on top of the primitive structures, mathematical knowledge can be incorporated by using decision procedures and constraint solvers as black boxes. This is the case, for instance, in constraint logic programming, in resolution theorem proving with theory unification [10, 11], or in theorem proving with constraints [19]. The case of sufficient-completely defined free functions has been investigated e.g. by Bachmair, Ganzinger, and Waldmann [6]; Avenhaus and Becker [1] have described a related order-sorted approach.

In an algebraic context, however, sufficient completeness practically excludes uninterpreted function symbols. A similar situation arises, for instance, with

* The research described in this paper was supported in part by the ESPRIT Basic Research Working Group 6028 (Construction of Computational Logics). We wish to thank Leo Bachmair, Deepak Kapur, David Plaisted, Georg Struth, and Jürgen Stuber for helpful discussions on the subject of this paper.

theorem provers based on extensions of Presburger arithmetic. Unable to handle existential quantification they are suited for verification (where instantiations are known), but not for proof discovery and program synthesis (Hines [16]). Furthermore, experiments by Boyer and Moore [9] show that mathematical routines rarely have a chance to contribute to a proof, unless they are tightly interwoven with the rest of the prover. Heuristic approaches that involve such a more sophisticated combination of modules, e.g., in the Boyer-Moore prover or in the PVS system [23], have proven to be very useful in practice.

Our own line of research is based on the integrating approach, where the inference rules are adapted to the theory in a specific way. Equality is the prime example for this technique: In paramodulation and superposition, there are no resolution inferences with the transitivity or symmetry axioms. Rather, the equality axioms are coded as inference rules, which can be subject to specific ordering restrictions. Other transitive relations, in particular orderings, can be handled by related rewriting techniques yielding chaining calculi as they have been investigated by Bledsoe, Kunen, and Shostak [8], Hines [15], and Bachmair and Ganzinger [3], among others. The laws of associativity and commutativity of binary functions have previously been integrated into paramodulation calculi: Extended clauses and unification modulo AC make explicit inferences with the AC axioms unnecessary (see Peterson and Stickel [24] for the case of unit equations, and Wertz [30] and Bachmair and Ganzinger [2] for the clausal case). Unfortunately, AC-unification is doubly exponential, and it shows in practice. By using constraints, AC-unification can be replaced by the "only" simply exponential problem of AC-unifiability, as described by Nieuwenhuis and Rubio [22] and Vigneron [29]. On the other hand, (some) equality constraints need to be solved for simplification. One of the results of the present paper is that there are other ways to avoid especially prolific cases of AC-unification in the presence of more algebraic structure.

The present paper describes a refined superposition calculus for cancellative abelian monoids. They encompass not only abelian groups, but also such ubiquitous structures as the natural numbers or multisets. The highlights of our calculus are the following. (i) There is no need for explicit inferences with the theory clauses for cancellative abelian monoids. Hence, there is no generation and recombination of different variants of one and the same clause. (ii) There is no need for extended equations or clauses. By virtue of this fact, many especially prolific instances of AC-unification are avoided. Actually, AC-unification can be replaced by ACU-unification. (iii) The ordering constraints for superposition can be further refined. As in previous calculi one may ignore non-maximal literals, as well as the smaller side of an equation for chaining. Here, in addition, only the maximal term of a (maximal) sum participates in an inference. (iv) The general notion of redundancy as it was introduced by Bachmair and Ganzinger [4] can be appropriately refined. It forms the basis for specific forms of the usual simplification techniques. Simplification is an indispensable component of any saturation-based theorem prover. In this paper, redundancy will in particular allow to show admissibility of certain variable elimination techniques.

A crucial indication of the practicality of any approach for integrating mathematical theories by specifically refined forms of resolution and paramodulation

or chaining is the extent to which it avoids a certain especially prolific form of inference. It is the type of inference in which the main term in one of the premises is a variable, and, hence, unification is no longer an effective filter. For the equational case it is known that paramodulation into or below variables is not needed. For dense, total orderings without endpoints similar results have been obtained in [3, 8, 15, 25]. In the calculus to be described below, inferences are computed with maximal terms in sums. The sum itself, obviously, is not a variable, but the term may be one. The fact that certain inferences involving such variables cannot be avoided is an indication for why non-refined superposition strategies can perform badly for monoids with cancellation. In our refined calculus, only unshielded variables, i.e., variables that do not occur somewhere else as arguments of free function symbols, pose a problem in this regard. That leaves us with the question of how to deal with unshielded variables. Fortunately unshielded variables can be eliminated in many, though not all cases without affecting refutational completeness.

With these characteristics, the present approach is a considerable improvement over Hsiang, Rusinowitch, and Sakai's extension of ordered paramodulation [17, 27] to handle cancellation laws. Our approach is related to normalized rewriting modulo the group axioms (Marché [20]) and superposition for integer modules (Stuber [28]). Both handle only the stronger case of groups. As will become apparent, working in "non-groups" makes some aspects of equational theorem proving significantly more difficult while others are simplified. For instance, there will sometimes be a need for abstraction[2]; in other situations, the number of overlaps is reduced, and in non-groups it may be easier to eliminate positive occurrences of unshielded variables.

In this extended abstract, we can only give an overview of the main ideas behind our work. Complete descriptions of the inference system and the corresponding completeness proof can be found in the full version [14].

## 2 The Cancellative Superposition Calculus

### 2.1 Preliminaries

**Definition 1.** An abelian semigroup is an algebraic structure consisting of a non-empty set $G$ and a binary function $+$ that satisfies the associativity axiom (A) and the commutativity axiom (C). An abelian monoid is an abelian semigroup $(G, +)$ with a distinguished constant $0 \in G$ for which the identity axiom (U) holds. An abelian semigroup $(G, +)$ or monoid $(G, +, 0)$ is called cancellative, if it satisfies additionally the cancellation axiom (K)

$$\forall x, y, z \in G: \qquad x + z = y + z \;\Rightarrow\; x = y \;\; .$$

We denote the set of the axioms (A), (C), (U), and (K) by ACUK, and the respective subsets by AC, ACU, and ACK. For $m \in \mathbf{N}^+$, $\sum_{i \in \{1,\ldots,m\}} x_i$ is an abbreviation for the sum $x_1 + \cdots + x_m$ and $mx$ is an abbreviation for the $m$-fold sum $x + \cdots + x$. In a monoid, $\sum_{i \in \emptyset} x_i$ and $0x$ are defined as 0.

[2] This would not even be possible in the purely equational framework of [20].

In a cancellative abelian semigroup or monoid, the difference between two elements is uniquely defined whenever it exists. It is a group, if and only if difference is a total function. Typical examples of cancellative abelian monoids where difference is partial are the natural numbers under addition, the non-zero integers under multiplication, and multisets under union.

We will develop our calculus in the framework of equational clauses. Terms are formed over a signature containing a binary function symbol $+$ and a constant symbol $0$.[3] An equation $e$ is an unordered pair $\{t, t'\}$ of terms, usually written as $t \approx t'$. A literal is either an equation $e$ (also called a positive literal) or a negated equation $\neg\, e$ (also called a negative literal). The symbol $[\neg]\, e$ denotes either of these. Instead of $\neg\, t \approx t'$, we sometimes write $t \not\approx t'$. A clause is a finite multiset of literals. We assume that the reader is familiar with standard concepts and notations in the area of rewriting (see Dershowitz and Jouannaud [12]), and in the area of superposition-based theorem proving (see Bachmair and Ganzinger [4]).

The symbol $=_{\mathrm{ACU}}$ denotes the congruence on terms generated by ACU. The ACU-congruence class of a term $t$ is $[t]_{\mathrm{ACU}} = \{\, t' \mid t =_{\mathrm{ACU}} t' \,\}$.

**Definition 2.** Every function symbol different from 0 and $+$ is called free. A term is called atomic, if it is not a variable and its top symbol is different from $+$. A term $t$ is called a proper sum, if $t = t_1 + t_2$ and $t_1 \neq_{\mathrm{ACU}} 0$, $t_2 \neq_{\mathrm{ACU}} 0$.

The set of all terms is the union of the three disjoint sets $\{\, t \mid \exists x\colon x \text{ is a variable}, t =_{\mathrm{ACU}} x \,\}$, $\{\, t \mid \exists s\colon s \text{ is atomic}, t =_{\mathrm{ACU}} s \,\}$, and $\{\, t \mid \exists s\colon s \text{ is a proper sum}, t =_{\mathrm{ACU}} s \,\}$. We can therefore extend the terminology above to ACU-congruence classes, and say that $[t]_{\mathrm{ACU}}$ is a variable (an atomic term, a proper sum), if there is some $s \in [t]_{\mathrm{ACU}}$ with this property.

We say that a term $t$ occurs at the top of $s$, if there is a position $o \in \mathrm{Pos}(s)$ such that $s|_o = t$ and for every proper prefix $o'$ of $o$, $s(o')$ equals $+$. We say that $t$ occurs in $s$ below a free function symbol, if there is an $o \in \mathrm{Pos}(s)$ such that $s|_o = t$ and $s(o')$ is a free function symbol for some proper prefix $o'$ of $o$. We extend this terminology to ACU-congruence classes, and say that $[t]_{\mathrm{ACU}}$ occurs in $[s]_{\mathrm{ACU}}$ at the top (below a free function symbol), if there are some $t' \in [t]_{\mathrm{ACU}}$ and $s' \in [s]_{\mathrm{ACU}}$ with this property. For instance, $[2a]_{\mathrm{ACU}}$ and $[a + f(a + b)]_{\mathrm{ACU}}$ occur at the top of $[3a + 2f(b + a)]_{\mathrm{ACU}}$, $[a]_{\mathrm{ACU}}$ occurs both at the top and below a free function symbol.

A reduction ordering $\succ$ is called ACU-compatible, if $s' =_{\mathrm{ACU}} s \succ t =_{\mathrm{ACU}} t'$ implies $s' \succ t'$. Every ACU-compatible reduction ordering induces naturally a reduction ordering on ACU-congruence classes. For ground terms, we can obtain an ACU-compatible reduction ordering $\succ$ from an arbitrary AC-compatible ordering $\succ_1$ by defining $s \succ t$ if $s{\downarrow} \succ_1 t{\downarrow}$, where $s{\downarrow}$ denotes the normal form of $s$ under rewriting with the rule $x + 0 \to x$. We can lift this ordering to non-ground terms by defining $s \succ t$ if $s\theta \succ t\theta$ for all ground instances $s\theta$ and $t\theta$. However, as shown by Jouannaud and Marché [18], it happens quite frequently that $\succ$ orders a pair of terms in an operationally undesirable way, or that $s[x]$ and $t[x]$ are un-

[3] There is no scalar multiplication in our signature, so $mt$ is merely an abbreviation for $t + \cdots + t$.

comparable because $s[0] \succ t[0]$ but $s[u] \prec t[u]$ for all non-zero ground terms $u$.[4] This is a serious problem, if one is interested in classical rewriting. It is not a hindrance, though, for calculi like superposition or unfailing completion, which are preferably implemented using constraints. (In fact, Jouannaud and Marché's method can be seen as a variant of unfailing completion with constraints.)

**Definition 3.** We say that an ACU-compatible ordering has the multiset property, if whenever a ground atomic term $u$ is greater than $v_i$ for every $i$ in a finite index set $I \neq \emptyset$, then $u \succ \sum_{i \in I} v_i$.

For the remainder of this paper, $\succ$ will always denote an ACU-compatible ordering that has the multiset property and is total on ground ACU-congruence classes.[5] Examples of orderings with these properties are obtained from Bachmair and Plaisted's APO [7] or the ordering of Rubio and Nieuwenhuis [26] with precedence $f_n \succ \ldots \succ f_1 \succ + \succ 0$ by comparing $s{\downarrow}$ and $t{\downarrow}$ as described above.

A positive ground literal $e$ is called true in a set $E$ of ground equations, if $e \in E$. A ground literal $\neg\, e$ is called true in $E$, if $e \notin E$. A ground clause is called true in $E$, if at least one of its literals is true in $E$. If a ground literal or clause is not true in $E$, it is called false in $E$. A set $E$ of ground equations is called a model of a clause $C$, if every ground instance $C\theta$ of $C$ is true in $E$; it is called a model of a set $N$ of clauses, if it is a model of every $C \in N$. If $N$ and $N'$ are sets of clauses, we write $N \models N'$ if every model of $N$, ACUK, and the equality axioms is a model of $N'$. In other words, $\models$ denotes entailment modulo ACUK.

**Convention 4.** From now on we will work *only* with ACU-congruence classes, rather than with terms. To simplify notation, we will omit the $[_]_{\text{ACU}}$ and drop the subscript of $=_{\text{ACU}}$. So *all* terms, equations, substitutions, inference rules, etc., are to be taken modulo ACU, i.e., as representatives of their congruence classes.

Our ordering restrictions are more refined than usual, in that they are based primarily on maximal terms in sums.

**Definition 5.** The maximal atomic term (with or without multiplicity) of a ground equation is defined as follows. Let $e$ be the equation $nu + \sum_{i \in I} s_i \approx mu + \sum_{j \in J} t_j$, where $u$, $s_i$, and $t_j$ are atomic terms, $n \geq m \geq 0$, $n \geq 1$, and $u \succ s_i$ and $u \succ t_j$ for all $i \in I$, $j \in J$. Then $\mathrm{mt}(e) = u$ and $\mathrm{mt}_{\#}(e) = nu$.

**Definition 6.** The symbol $\mathrm{ms}(t)$ denotes the multiset of all non-zero atomic terms occurring at the top of a term $t$, i.e., $\mathrm{ms}(t) = \{\, v_j \mid j \in J \,\}$, if $t = \sum_{j \in J} v_j$ and all $v_j$ are non-zero atomic terms. ($J$ may be empty, thus $\mathrm{ms}(0) = \emptyset$.) If $e$ is an equation $t \approx t'$, then $\mathrm{ms}(e)$ is the multiset union of $\mathrm{ms}(t)$ and $\mathrm{ms}(t')$.

The ordering $\succ$ on terms induces an ordering on literals as follows: Every ground literal $[\neg]\, s \approx t$ is mapped to the quadruple $(\mathrm{mt}_{\#}(s \approx t), \mathit{pol}, \mathrm{ms}(s \approx t), \{s, t\})$ where $\mathit{pol}$ is 1 for negative literals and 0 for positive ones. Two ground

[4] Jouannaud and Marché's statement that "AC1-rewrite orderings cannot really exist" [18] should be taken with a grain of salt, however.

[5] In practice, it is sufficient if the ordering can be extended to a total ordering.

literals are compared by comparing their associated quadruples using the lexicographic combination of the ordering $\succ$ on terms, the ordering $>$ on $\mathbf{N}$, the multiset extension of $\succ$, and the multiset extension of $\succ$. We denote this literal ordering also by $\succ$. It is lifted to possibly non-ground literals in the usual way, so $[\neg]\, e_1 \succ [\neg]\, e_2$ if and only if $[\neg]\, e_1\theta \succ [\neg]\, e_2\theta$ for all ground instances $[\neg]\, e_1\theta$ and $[\neg]\, e_2\theta$. In order to compare equations, they are identified with positive literals. The ordering $\succ$ on clauses is the multiset extension of the literal ordering $\succ$. As the orderings on literals and clauses are obtained from noetherian orderings by multiset extension and lexicographic combination, they are noetherian, too.

### 2.2 The Inference Rules

*The Superposition Calculus.* The superposition calculus of Bachmair and Ganzinger [4] is a refutationally complete theorem proving method for arbitrary first-order clauses with equality. Starting from an initial set of formulae, a superposition-based theorem prover computes inferences, e.g.,

$$\frac{D \vee s \approx s' \qquad C \vee [\neg]\, t[u] \approx t'}{(D \vee C \vee [\neg]\, t[s'] \approx t')\sigma} \qquad \sigma = \mathrm{mgu}(s, u)$$

and adds their conclusions to the set of formulae. If the initial set is inconsistent, then every fair derivation will eventually produce a contradiction (i.e., the empty clause). The inference rules are restricted forms of paramodulation, resolution, and factoring. For instance, it is sufficient to perform only those superposition inferences that involve maximal literals and maximal sides of equalities. Besides, a global redundancy criterion allows to discard certain inferences and formulae.

Our goal is to develop a refined and otherwise similar calculus for cancellative abelian monoids that makes superpositions with the ACUK axioms superfluous.

*Cancellative Superposition.* Let us first restrict to the case that $+$ is the only non-constant function symbol. If we have an equation $u + s \approx s'$ where the atomic term $u$ is larger than $s$ and $s'$, then we can deduce $[\neg]\, s' + t \approx s + t'$ from $[\neg]\, u + t \approx t'$ by congruence and cancellation. Intuitively, this means that rather than replacing the left hand side of a rewrite rule by the right hand side, we replace the maximal atomic part by the remainder: We rewrite $u$ to $s'$ while adding $s$ to the other side of the (possibly negated) equation. This is the essential reason why extended rules are unnecessary in our calculus.

The method can be generalized to equational clauses. Taking into account that $u$ might occur more than once in a sum we get the (ground) inference rule

$$\underline{\text{Cancellative Superposition}} \qquad \frac{D \vee mu + s \approx s' \quad C \vee [\neg]\, nu + t \approx t'}{D \vee C \vee [\neg]\, (n{-}m)u + t + s' \approx t' + s}$$

where $n \geq m \geq 1$.[6]

[6] Recall that we are working with terms modulo ACU. In particular, this implies that $s$ and $t$ may be missing (i.e., zero).

Together with the cancellation, equality resolution, and cancellative equality factoring rules, this rule is refutationally complete for sets of ground clauses, provided that $+$ is the only non-constant function symbol.

$$\text{Cancellation} \qquad \frac{D \vee [\neg]\, nu + t \approx mu + s}{D \vee [\neg]\, (n{-}m)u + t \approx s}$$

$$\text{Equality Resolution}^{7} \qquad \frac{C \vee \neg\, 0 \approx 0}{C}$$

$$\text{Cancellative Eq. Factoring} \qquad \frac{D \vee nu + s \approx s' \vee nu + t \approx t'}{D \vee \neg\, s + t' \approx s' + t \vee nu + t \approx t'}$$

The inference system remains refutationally complete if we add ordering restrictions, such that inferences are computed only if the literals involved are maximal in their clauses and $u$ is atomic and strictly larger than $s$, $s'$, $t$, and $t'$.

**Example 7.** Suppose that the ordering on constant symbols is given by $b \succ b' \succ c \succ d \succ d'$. We will show that the following four clauses are contradictory with respect to ACUK. (The maximal part(s) of every clause are underlined.)

$$\underline{2b} + c \approx d \tag{1}$$
$$\underline{b'} + c \approx d' \tag{2}$$
$$\underline{d} \approx d' \tag{3}$$
$$\underline{2b} \not\approx b' \tag{4}$$

Cancellative superposition of (1) and (4) yields

$$d \not\approx \underline{b'} + c \tag{5}$$

Cancellative superposition of (2) and (5) yields

$$d + \underline{c} \not\approx d' + \underline{c} \tag{6}$$

By cancellation of (6) we obtain

$$\underline{d} \not\approx d' \tag{7}$$

Cancellative superposition of (3) and (7) produces

$$\underline{d'} \not\approx \underline{d'} \tag{8}$$

which by cancellation and equality resolution yields the empty clause.

Speaking in terms of AG-normalized completion (Marché [20]), we can work directly with the symmetrisation; Marché's $\Psi_{\mathrm{AG}}$ and $\Theta_{\mathrm{AG}}$ have no counterpart in our framework. Consequently, the number of overlaps that have to be considered is reduced. On the other hand, we lack an inverse, which will lead to certain problems once free function symbols are introduced.

[7] As the cancellation rule transforms $C \vee \neg\, s \approx s$ into $C \vee \neg\, 0 \approx 0$, it suffices to handle only the latter by equality resolution.

*The Non-Ground Case.* To lift the standard superposition calculus to non-ground clauses, one simply needs to replace equality in the ground inference by unifiability (or by equality constraints). The situation is similar here, as long as all variables in our clauses are shielded:

**Definition 8.** Let $x$ be a variable occurring in some literal or clause. We say that $x$ is shielded in the literal or clause, if it occurs at least once below a free function symbol. Otherwise, $x$ is called unshielded.

For example, both $x$ and $z$ are shielded in $x + y \approx g(z) \vee f(x) + z \approx 0$, whereas $y$ is unshielded. Unshielded variables are operationally difficult to handle as they may correspond to maximal atomic subterms in a ground instance. For shielded variables, this is impossible: If $x$ is shielded in a clause, then the clause contains an atomic subterm $t[x]$. As $x\theta \prec (t[x])\theta$, an atomic subterm of $x\theta$ cannot be maximal.

In the subsequent definition of cancellative superposition for non-ground clauses, we assume that all variables in the clauses are shielded. Cancellation and cancellative equality factoring inferences are lifted in a similar way.

$$\underline{\text{Cancellative Superposition}} \quad \frac{D \vee e_1 \quad C \vee [\neg]\, e_2}{(D \vee C \vee [\neg]\, e_0)\sigma}$$

if the following conditions are satisfied:[8]

- $e_1 \;=\; \sum_{k\in K} m_k u_k + s \approx s'$.
- $e_2 \;=\; \sum_{l\in L} n_l v_l + t \approx t'$.
- All $u_k$ and $v_l$ are atomic terms (in particular not variables).
- $1 \le m = \sum_{k\in K} m_k \le n = \sum_{l\in L} n_l$.
- $e_0 \;=\; (n{-}m)z + t + s' \approx s + t'$.
- $\sigma$ is a most general ACU-unifier of all $u_k$, $v_l$, and $z$ ($k \in K$, $l \in L$).
- $e_1\sigma$ is a strictly maximal literal in $(D \vee e_1)\sigma$, $[\neg]\, e_2\sigma$ is a maximal literal in $(C \vee [\neg]\, e_2)\sigma$ (strictly maximal, if $e_2$ occurs positively).
- $(D \vee e_1)\sigma \not\succeq (C \vee [\neg]\, e_2)\sigma$, $z\sigma \not\preceq s\sigma$, $z\sigma \not\preceq s'\sigma$, $z\sigma \not\preceq t\sigma$, $z\sigma \not\preceq t'\sigma$.

Lifting the calculus to non-ground clauses becomes a non-trivial task, if unshielded variables have to be taken into account. It may happen that an unshielded variable is instantiated with a sum and that only the maximal atomic part of the sum takes part in the unification. We have shown in the full version [14] of this paper that lifting is still possible in this case. The inference rules become more complicated, but still produce only finitely many conclusions. By lack of space, we omit them here. As the inference rules become also more prolific, inferences involving unshielded variables should in practice be avoided whenever possible. In Sect. 4 we will discuss suitable techniques to do this.

[8] Many of the conditions should preferably be implemented by constraints and inherited from ancestor clauses (with the usual caveats concerning simplification, cf. Nieuwenhuis and Rubio [21])

*Free Function Symbols.* As soon as the clauses contain non-constant free function symbols, we also have to use the inference rules of the traditional superposition calculus, i. e., equality resolution, standard superposition, and standard equality factoring. But this is not sufficient, as shown by the following example.

**Example 9.** Let the ordering on constants be given by $b \succ b' \succ c \succ d \succ d'$. In every equality model of ACUK and the clauses (1)–(3), the terms $2b$ and $b'$ are equal.

$$\underline{2b} + c \approx d \tag{1}$$

$$\underline{b'} + c \approx d' \tag{2}$$

$$\underline{d} \approx d' \tag{3}$$

As we have shown in Example 7 we can thus refute the set of clauses (1)–(4).

$$\underline{2b} \not\approx b' \tag{4}$$

There is no way, however, to perform a cancellative superposition inference below a free function symbol, hence there is no way to derive the empty clause from the clauses (1), (2), (3), and (9).

$$\underline{f(2b)} \not\approx f(b') \tag{9}$$

If we were working in groups, we could simply derive $f(d - c) \not\approx f(b')$. But this is impossible in our context. Hsiang, Rusinowitch, and Sakai [17, 27] have solved this problem by introducing the inference rule

$$\underline{\text{C2}} \qquad \frac{D \vee u + s \approx s' \quad C \vee v + s \approx s'}{D \vee C \vee u \approx v}$$

In the example above, this rule allows to derive $\underline{2b} \approx b'$ from the first three clauses, which can then be applied to (9) by standard superposition. However, there is a drawback of this approach. Before we can apply the C2 rule, we have to use clause (3) to replace $d$ by $d'$ in (1). Since the term $d$ is not maximal in (1), the C2 rule can be only used in conjunction with ordered paramodulation (where inferences may involve *smaller* parts of maximal literals), but does not work together with strict superposition (where such inferences are excluded).

The concept of abstraction yields another solution for the problem, which fits more smoothly into the superposition calculus. Abstracting out an occurrence of a term $v$ in a clause $C[v]$ means replacing $v$ by a new variable $x$ and adding $x \not\approx v$ as a new condition to the clause. In our case, we have to abstract out a term $v$ occurring immediately below a free function symbol, if there is some other clause $D \vee mu + s \approx s'$ such that (i) $mu$ occurs at the top of $v$, but (ii) a standard superposition of $mu + s$ into $v$ is impossible. We emphasize that the new variable $x$ is always shielded in the resulting clause.

$$\underline{\text{Abstraction}} \qquad \frac{C \vee [\neg]\, w[nu + t] \approx w'}{C \vee \neg\, x \approx nu + t \vee [\neg]\, w[x] \approx w'}$$

where $n \geq m$.

**Example 10.** Using the abstraction rule, the set of clauses (1), (2), (3), and (9) of Example 9 can be refuted as follows: Abstraction of (1) and (9) yields

$$x \not\approx 2b \vee \underline{f(x)} \not\approx \underline{f(b')} \tag{10}$$

By (non-ground) cancellation of (10) with the unifier $\{x \mapsto b'\}$ we obtain

$$b' \not\approx \underline{2b} \vee 0 \not\approx 0 \tag{11}$$

which can be refuted in the same way as (4) in Example 7.

The abstraction rule is extended to non-ground premises in the same way as the cancellative superposition rule.

### 2.3 Redundancy

To make a saturation-based theorem proving technique practically useful, the inference system has to be complemented with a redundancy criterion. Given a set $N$ of clauses, a clause is redundant with respect to $N$, if it follows from the equality and ACUK axioms and smaller clauses in $N$. It can be deleted from the current set of clauses at any point of the saturation process. An inference is redundant, if its conclusion follows from from the equality and ACUK axioms and clauses in $N$ that are smaller than the largest premise. It may be ignored during the saturation process without endangering the fairness of the derivation. (See [4, 6] for a more detailed discussion of these aspects of superposition. Note that "redundancy" is called "compositeness" in [4].)

## 3 Refutational Completeness

### 3.1 Rewriting on Equations

To prove that the inference system described so far is refutationally complete we have to show that every saturated clause set that doesn't contain the empty clause has a model. The traditional approach to construct such a model is rewrite-based: First an ordering is imposed on the set of all ground instances of clauses in the set. Starting with an empty interpretation all such instances are inspected in ascending order. If a reductive clause is false and irreducible in the partial interpretation constructed so far, its maximal equation is turned into a rewrite rule and added to the interpretation. If the original clause set is saturated and doesn't contain the empty clause, then the final interpretation is a model of all ground instances, and thus of the original clause set (Bachmair and Ganzinger [4]). In our case, we have to modify this scheme. For an adequate treatment of cancellative superposition it is not sufficient to be able to replace equals by equals inside a term. Rather, we need a rewrite relation on equations.

**Definition 11.** A ground equation $e$ is called a rewrite rule if $\mathrm{mt}(e)$ doesn't occur on both sides of $e$.

Every rewrite rule has the form $nv + s \approx s'$, where $v$ is an atomic term, $n \in \mathbf{N}^+$, $v \succ s$, and $v \succ s'$. This follows from the multiset property of $\succ$.

At the top of a term, we will use rewrite rules in a specific way: Application of a rule $nv + s \approx s'$ to an equation $nv + t \approx t'$ means to replace $nv$ by $s'$ and

simultaneously to add $s$ to the other side, obtaining $s' + t \approx t' + s$.[9] However, this is only possible at the top of an equation, not below a free symbol. Consequently, there may be equations $t \approx t'$ that can be reduced to $0 \approx 0$, whereas $f(t) \approx f(t')$ is irreducible. To compensate for this fact our rewrite relation takes two sets of rewrite rules as parameters: one set of rules generated directly from the clause set, and one set of "derived" rules, which are applied only below a free symbol and will fix the above problem.

**Definition 12.** Given a pair $(R, R')$ of sets of rewrite rules, the three relations $\to_{\gamma,(R,R')}$, $\to_{\delta,(R,R')}$, and $\to_{\kappa}$ are defined (modulo ACU) as follows:

- $nv + t \approx t' \;\to_{\gamma,(R,R')}\; s' + t \approx t' + s$,
  if $nv + s \approx s'$ is a rule in $R$.
- $t[s] \approx t' \;\to_{\delta,(R,R')}\; t[s'] \approx t'$,
  if $s \approx s'$ is a rule in $R \cup R'$ and $s$ occurs in $t$ below some free function symbol.
- $u + t \approx u + t' \;\to_{\kappa}\; t \approx t'$,
  if $u$ is different from 0.

The union of $\to_{\gamma,(R,R')}$, $\to_{\delta,(R,R')}$, and $\to_{\kappa}$ is denoted by $\to_{(R,R')}$.

The relation $\to_{(R,R')}$ is contained in the literal ordering $\succ$ and thus noetherian. We say that an equation $e$ is $\gamma$-reducible, if $e \to_{\gamma} e'$ (analogously for $\delta$ and $\kappa$). It is called reducible, if it is $\gamma$-, $\delta$- or $\kappa$-reducible. Unlike $\kappa$-reducibility, $\gamma$- and $\delta$-reducibility can be extended to terms: A term $t$ is called $\gamma$-reducible, if $t \approx t' \to_{\gamma} e'$, where the rewrite step takes place at the left hand side (analogously for $\delta$). It is called reducible, if it is $\gamma$- or $\delta$-reducible.

**Definition 13.** The truth set $\mathrm{tr}(R, R')$ of a pair $(R, R')$ of sets of rewrite rules is the set of all equations $e$ for which there exists a derivation $e \;\to^*_{(R,R')}\; 0 \approx 0$.

### 3.2 Model Construction

A ground clause $C \vee e$ is called reductive for $e$, if $e$ is a rewrite rule and strictly maximal in $C \vee e$.

Let $N$ be a set of (possibly non-ground) clauses and let $\overline{N}$ be the set of all ground instances of clauses in $N$. Using induction on the clause ordering we define sets of rules $R_C$, $R_C^{\mathrm{K}}$, $E_C$, and $E_C^{\mathrm{K}}$, for all clauses $C \in \overline{N}$. Let $C$ be such a clause and assume that $R_D$, $R_D^{\mathrm{K}}$, $E_D$, and $E_D^{\mathrm{K}}$ have already been defined for all $D \in \overline{N}$ such that $C \succ D$. Then $R_C = \bigcup_{D \prec C} E_D$ and $R_C^{\mathrm{K}} = \bigcup_{D \prec C} E_D^{\mathrm{K}}$.

$E_C$ is the singleton set $\{e\}$, if $C$ is a clause $C' \vee e$ such that (i) $C$ is reductive for $e$, (ii) $C$ is false in $\mathrm{tr}(R_C, R_C^{\mathrm{K}})$, (iii) $C'$ is false in $\mathrm{tr}(R_C \cup \{e\}, R_C^{\mathrm{K}})$, and (iv) $\mathrm{mt}_{\#}(e)$ is irreducible with respect to $(R_C, R_C^{\mathrm{K}})$. Otherwise, $E_C$ is empty.

The set $E_C^{\mathrm{K}}$ is non-empty only if $E_C$ contains a rule $nv + s \approx s'$. In this case, $E_C^{\mathrm{K}}$ is the set of all $knv + r \approx r' \in \mathrm{tr}(R_C \cup E_C, R_C^{\mathrm{K}})$ such that $v \succ r$, $v \succ r'$, and $knv + r \approx r'$ is $\delta\kappa$-irreducible with respect to $(R_C, R_C^{\mathrm{K}})$.

Finally, we define $R_\infty = \bigcup_{D \in \overline{N}} E_D$ and $R_\infty^{\mathrm{K}} = \bigcup_{D \in \overline{N}} E_D^{\mathrm{K}}$.

[9] While we have the restriction $v \succ s$, $v \succ s'$ for the rewrite rules, there is no such restriction for the equations to which rules are applied.

**Theorem 14.** *For every* $C \in \overline{N}$, $\mathrm{tr}(R_C, R_C^{\mathrm{K}})$ *and* $\mathrm{tr}(R_\infty, R_\infty^{\mathrm{K}})$ *are models of the equality axioms and* ACUK.

**Theorem 15.** *Let* $N$ *be a set of clauses that is saturated up to* ACUK*-redundancy and does not contain the empty clause. Then* $\mathrm{tr}(R_\infty, R_\infty^{\mathrm{K}})$ *is a model of every ground clause* $C \in \overline{N}$.

**Corollary 16.** *Let* $N$ *be a set of clauses that is saturated up to* ACUK*-redundancy. Then* $N \cup \mathrm{ACUK}$ *is equality unsatisfiable if and only if* $N$ *contains the empty clause.*

## 4 Simplification Techniques

**Definition 17.** The ordering $\succ_{\mathrm{x}}$ on ground terms is defined as follows: $s \succ_{\mathrm{x}} t$ if there is an $s' \in \mathrm{ms}(s)$ such that $s' \succ t'$ for all $t' \in \mathrm{ms}(t)$. For arbitrary terms, $s \succ_{\mathrm{x}} t$ if $s\theta \succ_{\mathrm{x}} t\theta$ for all ground instances $s\theta$ and $t\theta$.

Let $N$ be a set of clauses. We say that $M \subseteq N$ is simplified to another set of clauses $M'$, if $N \models M'$ (so that we may add $M'$, once we have got $N$), and if $M$ is ACUK-redundant with respect to $N \cup M'$ (so that we may delete $M$, once we have added $M'$). For example, every clause $C_1 \;=\; C \vee [\neg]\, s + t \approx s + t'$ can be simplified to $C_0 \;=\; C \vee [\neg]\, t \approx t'$ (independently of $N$). We can also extend the classical simplification by demodulation to cancellative superposition, so that a clause $C_1 \;=\; C \vee C' \vee [\neg]\, s + t \approx t'$ can be simplified to $C_0 \;=\; C \vee C' \vee [\neg]\, w' + t \approx t' + w$ provided that $N$ contains a clause $D$ such that $D\sigma \;=\; C' \vee s + w \approx w'$ where $s \succ_{\mathrm{x}} w$, $s \succ_{\mathrm{x}} w'$, and $C_1 \succ D\sigma$. For instance, every inference in Example 7 is a simplification of the maximal premise. In particular, this technique can in nearly all cases be used to eliminate any remaining occurrences of the redex in the conclusion of a cancellative superposition inference.

Bachmair and Ganzinger [4] list a number of general simplification techniques such as case analysis or contextual rewriting. These methods can easily be extended to our framework. In this section, we will concentrate on techniques that help to reduce the number of clauses with unshielded variables.

Whereas the ordering restrictions of our inference rules make inferences into shielded variables superfluous, cancellative superposition inferences into unshielded variables cannot generally be avoided. As an example, consider the clauses $b + c \approx d$ and $x + c \not\approx d$ with the ordering $b \succ c \succ d$. As unification is not an effective filter, clauses with unshielded variables are extremely prolific. Simplification techniques allow to remove certain clauses with unshielded variables from the clause set, or at least to render inferences with them redundant.

**Lemma 18.** *Let* $(G, +)$ *be a cancellative abelian semigroup and* $M$ *be a finite subset of* $G$. *If for every* $x \in G$ *there exists a* $y \in G$ *such that* $x + y \in M$, *then* $G$ *is a group.*

**Corollary 19.** *Let* $C$ *be a clause* $C' \vee \bigvee_{i \in I} n_i x + s_i \approx t_i$ *where* $x$ *occurs neither in* $C'$ *nor in the* $t_i$. *Let* $N$ *be a set of clauses and* $0$ *and* $-$ *be new function symbols. Then every model of* $\{C\} \cup \mathrm{ACK}$ *is either a model of* $C'$ *or can be extended to a*

*model of* $x+0 \approx x$ *and* $x+(-x) \approx 0$. *In particular,* $N \cup \{C\} \cup \mathrm{ACK}$ *is satisfiable if and only if* $N \cup \{C'\} \cup \mathrm{ACK}$ *or* $N \cup \{C\} \cup \{x+(-x) \approx 0\} \cup \mathrm{ACUK}$ *is.*

In its most general form, we can use this corollary to split one theorem proving derivation into two branches in a tableaux-like manner (cf. [5]). It is particularly useful if one of the two branches fails immediately. This happens in two situations: The first branch can be closed, if $C'$ is empty. In this case $C$ implies the identity and inverse axioms, and, although it is not required by fairness, it may be wise to add them to find a shorter proof.[10] The second branch can be closed, if $N$ contains some subset $N'$ that implies that the model is not a group. (For instance, $N'$ might consist of the single clause $y + a \not\approx b$.) In this case, $C$ can be simplified to $C'$. In "non-groups" it is thus always possible to get rid of unshielded variables that occur only positively.

Unshielded variables occurring negatively are harder to handle. There is a variant of "rewriting with equations of conditions" which can sometimes be applied if an unshielded variable occurs in more than one literal. A clause $C_1 = C \vee \neg\, mx + s \approx s' \vee [\neg]\, nx + t \approx t'$ with $n \geq m \geq 1$ is equivalent to $C_0 = C \vee \neg\, mx + s \approx s' \vee [\neg]\, (n-m)x + t + s' \approx t' + s$. Repeated use of this inference leads to a clause in which $x$ occurs only in one negative literal $\neg\, kx + w \approx w'$ and possibly in positive literals (with coefficients smaller than $k$). Unfortunately, this is not a simplification for all instances $C_1\theta$, but just for those that satisfy $x\theta \succ_{\mathrm{x}} s\theta$ and $x\theta \succ_{\mathrm{x}} s'\theta$. Hence adding $C_0$ makes inferences with $C_1$ unnecessary, if they involve only $x$ but no subterm of $s$ or $s'$. Inferences that involve both $x$ and a subterm of $s$ or $s'$ are still necessary, though.

If an unshielded variable occurs only in one negative literal (and no positive one), we can eliminate it, provided that the coefficient of the variable is 1 and that the inverse axiom has been derived: In a group, every clause of the form $C \vee \neg\, x + s \approx s'$, where $x$ doesn't occur in $C$, can be simplified to $C$. To eliminate unshielded variables that occur negatively with a coefficient $k$ larger than 1, and possibly also positively with coefficients smaller than $k$, additional properties of the model class are required, namely, divisibility and torsionfreeness. We are currently investigating to which extent this can be exploited.

## 5 Conclusions

We have presented a calculus for first-order equational theorem proving in the presence of the axioms of cancellative abelian monoids. The calculus is refutationally complete without requiring extended clauses or explicit inferences with the theory clauses. Compared with the conventional superposition calculus, on which it is based, the ordering constraints are strengthened in such a way that we may not only restrict to inferences that involve the maximal side of the maximal literal, but even to inferences that involve the maximal atomic terms. As we have shown in [14], the calculus may further be furnished with selection functions.

In traditional AC-superposition, extended rules show a rather prolific behaviour. In our approach, cancellative superposition makes extended rules su-

[10] The reader is invited to try to refute $\mathrm{ACUK} \cup \{\, 2x \approx a \vee x \approx b,\ 2b \not\approx a \,\}$ without first deducing the inverse axiom.

perfluous, and the ordering constraints mentioned above allow to exclude inferences involving shielded variables altogether. Many occurrences of unshielded variables can be eliminated by appropriate simplification techniques. Unfortunately cancellative superpositions into variables in sums cannot be completely avoided. More ways to eliminate unshielded variables are possible in the presence of further algebraic structure. This is a matter of further investigation.

At the time of writing this abstract, we cannot yet report about practical experiences with our calculus. An implementation in the Saturate system [13] is under way.

## References


1. Jürgen Avenhaus and Klaus Becker. Conditional rewriting modulo a built-in algebra. SEKI Report SR-92-11, Fachbereich Informatik, Univ. Kaiserslautern, 1992.
2. Leo Bachmair and Harald Ganzinger. Associative-commutative superposition. In Nachum Dershowitz and Naomi Lindenstrauss, eds., *Conditional and Typed Rewrite Systems, 4th International Workshop, CTRS-94*, Jerusalem, Israel, July 1994, LNCS 968, pp. 1–14. Springer-Verlag.
3. Leo Bachmair and Harald Ganzinger. Ordered chaining for total orderings. In Alan Bundy, ed., *Twelfth International Conference on Automated Deduction*, Nancy, France, June/July 1994, LNAI 814, pp. 435–450. Springer-Verlag.
4. Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.
5. Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Superposition with simplification as a decision procedure for the monadic class with equality. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, eds., *Computational Logic and Proof Theory, Third Kurt Gödel Colloquium*, Brno, Czech Republic, August 1993, LNCS 713, pp. 83–96. Springer-Verlag.
6. Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Refutational theorem proving for hierarchic first-order theories. *Applicable Algebra in Engineering, Communication and Computing*, 5(3/4):193–212, April 1994.
7. Leo Bachmair and David A. Plaisted. Termination orderings for associative-commutative rewriting systems. *Journal of Symbolic Computation*, 1:329–349, 1985.
8. W. W. Bledsoe, K. Kunen, and R. Shostak. Completeness results for inequality provers. *Artificial Intelligence*, 27:255–288, 1985.
9. Robert S. Boyer and J Strother Moore. Integrating decision procedures into heuristic theorem provers: A case study of linear arithmetic. In Jean E. Hayes, Donald Michie, and Judith Richards, eds., *Machine Intelligence 11: Logic and the acquisition of knowledge*, ch. 5, pp. 83–124. Oxford University Press, 1988.
10. Hans-Jürgen Bürckert. A resolution principle for clauses with constraints. In Mark E. Stickel, ed., *10th International Conference on Automated Deduction*, Kaiserslautern, FRG, July 1990, LNAI 449, pp. 178–192. Springer-Verlag.
11. Hans-Jürgen Bürckert. *A Resolution Principle for a Logic with Restricted Quantifiers*. LNAI 568. Springer-Verlag, Berlin, Heidelberg, New York, 1991.
12. Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, ed., *Handbook of Theoretical Computer Science*, vol. B: Formal Models and Semantics, ch. 6, pp. 243–320. Elsevier Science Publishers B.V., 1990.
13. Harald Ganzinger and Robert Nieuwenhuis. The Saturate system. URL: http://www.mpi-sb.mpg.de/SATURATE/Saturate.html, 1994.

14. Harald Ganzinger and Uwe Waldmann. Theorem proving in cancellative abelian monoids. Technical Report MPI-I-96-2-001, Max-Planck-Institut für Informatik, Saarbrücken, January 1996. URL: ftp://ftp.mpi-sb.mpg.de/pub/guide/staff/uwe/paper/MPI-I-96-2-001.ps.gz.
15. Larry M. Hines. Completeness of a prover for dense linear logics. *Journal of Automated Reasoning*, 8:45–75, 1992.
16. Larry M. Hines. Str+ve and integers. In Alan Bundy, ed., *Twelfth International Conference on Automated Deduction*, Nancy, France, June/July 1994, LNAI 814, pp. 416–430. Springer-Verlag.
17. Jieh Hsiang, Michaël Rusinowitch, and Ko Sakai. Complete inference rules for the cancellation laws (extended abstract). In John McDermott, ed., *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, vol. 2, pp. 990–992. Morgan Kaufmann Publishers, Inc.
18. Jean-Pierre Jouannaud and Claude Marché. Termination and completion modulo associativity, commutativity and identity. *Theoretical Computer Science*, 104(1):29–51, October 1992.
19. Claude Kirchner, Hélène Kirchner, and Michaël Rusinowitch. Deduction with symbolic constraints. *Revue Française d'Intelligence Artificielle*, 4(3):9–52, 1990.
20. Claude Marché. Normalised narrowing and normalised completion. In *Ninth Annual IEEE Symposium on Logic in Computer Science*, Paris, France, July 1994, pp. 394–403. IEEE Computer Society Press.
21. Robert Nieuwenhuis and Albert Rubio. Theorem proving with ordering constrained clauses. In Deepak Kapur, ed., *11th International Conference on Automated Deduction*, Saratoga Springs, NY, USA, June 1992, LNAI 607, pp. 477–491. Springer-Verlag.
22. Robert Nieuwenhuis and Albert Rubio. AC-superposition with constraints: no AC-unifiers needed. In Alan Bundy, ed., *Twelfth International Conference on Automated Deduction*, Nancy, France, June/July 1994, LNAI 814, pp. 545–559. Springer-Verlag.
23. S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, ed., *11th International Conference on Automated Deduction*, Saratoga Springs, NY, USA, June 1992, LNAI 607, pp. 748–752. Springer-Verlag.
24. Gerald E. Peterson and Mark E. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28(2):233–264, April 1981.
25. M. M. Richter. Some reordering properties for inequality proof trees. In E. Börger, G. Hasenjäger, and D. Rödding, eds., *Logic and Machines: Decision Problems and Complexity*, 1984, LNCS 171, pp. 183–197. Springer-Verlag.
26. Albert Rubio and Robert Nieuwenhuis. A precedence-based total AC-compatible ordering. In Claude Kirchner, ed., *Rewriting Techniques and Applications, 5th International Conference, RTA-93*, Montreal, Canada, June 1993, LNCS 690, pp. 374–388. Springer-Verlag.
27. Michaël Rusinowitch. *Démonstration Automatique: Techniques de Réécriture*, ch. 7: Ensembles complets de règles d'inférence pour les axiomes de régularité, pp. 111–127. InterEditions, Paris, 1989.
28. Jürgen Stuber. Superposition theorem proving for abelian groups represented as integer modules. To appear in Proc. RTA'96, 1996.
29. Laurent Vigneron. Associative-commutative deduction with constraints. In Alan Bundy, ed., *Twelfth International Conference on Automated Deduction*, Nancy, France, June/July 1994, LNAI 814, pp. 530–544. Springer-Verlag.
30. Ulrich Wertz. First-order theorem proving modulo equations. Technical Report MPI-I-92-216, Max-Planck-Institut für Informatik, Saarbrücken, April 1992.