Superposition modulo a Shostak Theory

Harald Ganzinger, Thomas Hillenbrand, and Uwe Waldmann

Max-Planck-Institut für Informatik, D-66123 Saarbrücken
{hg,hillen,uwe}@mpi-sb.mpg.de

Abstract. We investigate superposition modulo a Shostak theory T in order to facilitate reasoning in the amalgamation of T and a free theory F. Free operators occur naturally for instance in program verification problems when abstracting over subroutines. If their behaviour in addition can be specified axiomatically, much more of the program semantics can be captured. Combining the Shostak-style components for deciding the clausal validity problem with the ordering and saturation techniques developed for equational reasoning, we derive a refutationally complete calculus on mixed ground clauses which result for example from CNF transforming arbitrary universally quantified formulae. The calculus works modulo a Shostak theory in the sense that it operates on canonizer normalforms. For the Shostak solvers that we study, coherence comes for free: no coherence pairs need to be considered.

1 Introduction

Deduction in the combination of built-in and free theories has been an important topic of research for more than twenty years. One main motivation stems from program analysis, where one has to reason about standard theories like numbers as well as about free function symbols that occur naturally when abstracting over subroutines.

Deduction methods for combined theories can be roughly split into two groups: On the one hand there are white-box approaches, where theory knowledge is tightly integrated into individual inference rules of the general deductive system. This has been investigated for numerous algebraic theories, such as AC ([Plo72], among others), AC1 (e. g., [JM92]), cancellative Abelian monoids [Wal02], or Abelian groups [GN00]. While this kind of combination offers great flexibility, the requirements on termination orderings, the mathematical effort to prove the completeness of any single combined system and the technical effort to implement it are considerable. On the other hand, in black-box approaches a theory module is linked in a modular fashion to a general deductive system and the interaction between both is limited to a relatively small interface. Examples include theory resolution [Sti85], constraint resolution [Bür90], hierarchic superposition [BGW94], and the combination procedures of Nelson and Oppen [NO79] and Shostak [Sho84].

Shostak introduced a congruence closure procedure that can be combined with decision procedures for other theories, provided that these theories have canonizers and solvers. A solver is essentially a unitary unification algorithm, that transforms an equation $s \simeq t$ either into an equivalent set of equations with variables on the left-hand side, or into $\perp$ (if the equation is unsatisfiable). A canonizer is a procedure that transforms every term into some normal form with respect to the given theory. In this paper, we use these main ingredients of Shostak's method to integrate a canonizable and solvable theory into a refutationally complete theorem proving calculus for equational first-order clauses: the superposition calculus of Bachmair and Ganzinger [BG94].

From the Shostak point of view, our calculus can be seen as an extension from a calculus for (dis-)equations to a calculus working on arbitrary clauses over mixed terms. From the superposition point of view, the canonizer and the solver become simplification devices that allow us to replace theory equality by syntactic equality and to deal efficiently with overlaps between theory axioms and other axioms. The effect is thus similar to the symmetrization technique used in theory completion and theory superposition calculi [LC86,Mar94,Stu00]. However, while symmetrization tries to make coherence pairs between theory axioms and other axioms convergent by adding additional rules, the canonizer and the solver transform equations in such a form that coherence pairs become trivial.

To capture the behaviour of defined functions in applications like program verification, it is of course desirable that a calculus is not limited to ground formulas (corresponding to the universal fragment), but works also on non-ground formulas. In order to keep the technical contents manageable, we restrict ourselves mainly to the ground case in this paper. Still, extending the calculus to non-ground clauses is possible to some extent, at least if variables occur only below free function symbols. This is considerably simplified by the fact that splitting disjunctions into individual formulas in a tableau-like manner is allowed (and for ground literals often useful), but not required, so that inferences between formulas remain local. Variables occurring below theory symbols are critical, however. Here, the limits of a pure black-box approach are reached, and further information on the internals of the canonizer and the solver is required in order to keep the inferences finitely branching.

In this paper we can only give an overview of the main ideas behind our work. For a more detailed treatment, please see the full version [GHW03].

2 Preliminaries

For most logical notions and notations, we refer to [NR01]. In particular, we work in a logic with built-in equality and assume without loss of generality that equality is the only predicate symbol. For simplicity of notation the equality symbol $\simeq$ is supposed to be symmetric. A literal $s \bowtie t$ is either an equation $s \simeq t$ or a disequation $s \not\simeq t$. A clause is a disjunction of literals; a Horn clause is a clause with at most one equation.

We consider two disjoint first-order signatures Δ and Φ of *theory function symbols* and *free function symbols*, respectively. Terms over Δ and the set of variables $\{x, y, \ldots\}$ are called *theory terms*. Every term u can be written as $u[s_1, \ldots, s_m]_{p_1,\ldots,p_m} \equiv u[\vec{s}]_{\vec{p}}$ where the s_i are the maximal non-theory subterms, i.e. every s_i has a free top symbol and any of its superterms has a theory top symbol. Note that $\vec{s}$ may be empty, and $\vec{p}$ may be the root position. This presentation is unique up to arrangement. If u has a theory top symbol, then we use the notation $u[\![\vec{s}]\!]_{\vec{p}}$ in that situation. As usually, positions will be omitted whenever they are not relevant. A term headed by a free symbol will be called a *Φ-top term* and denoted as $f(\vec{s})$. Correspondingly $u[\![\vec{s}]\!]$ is called a *Δ-top term*. The syntactic equality of two terms s and t is denoted by $s \equiv t$.

The semantics of the theory symbols is given by a Δ-theory $\mathcal{T}$, i. e. by a class of Δ-structures that is closed under isomorphisms. We assume that $\mathcal{T}$ is convex and contains all its σ-models $\sigma(\mathcal{T})$ in the sense of [RS01], that is, all Δ-structures $\mathcal{M}$ such that (i) $\mathcal{M} \models \forall X.s \simeq t$ whenever $s =_{\mathcal{T}} t$, and (ii) $\mathcal{M} \models s \not\simeq t$ for $s \neq_{\mathcal{T}} t$ where s and t are ground. The theory models are considered in contexts where additional free functions from Φ exist. To that end, by $\mathcal{T}^{\Phi}$ we denote the class of those $\Delta \cup \Phi$-structures the restriction of which to Δ is in $\mathcal{T}$. As an abbreviation for $\mathcal{T} \models \forall X.s \simeq t$, where all the free variables of s and t are bound, we simply write $s =_{\mathcal{T}} t$. Correspondingly $s \neq_{\mathcal{T}} t$ denotes $\mathcal{T} \models \forall X.s \not\simeq t$.

3 Basic components

3.1 The canonizer and its extension

Every Shostak theory $\mathcal{T}$ comes with a so-called canonizer, a special simplification device that decides the word problem of $\mathcal{T}$. Following [KC03], we briefly recall the formalization of this concept and describe how it can be used to decide the word problem for the extension of $\mathcal{T}$ with free function symbols.

Assumption 1. The canonizer σ is required to fulfill the following properties:
(i) soundness: $t =_{\mathcal{T}} \sigma(t)$ (ii) completeness: $s =_{\mathcal{T}} t$ implies $\sigma(s) \equiv \sigma(t)$
(iii) idempotence: $\sigma^2(t) \equiv \sigma(t)$ (iv) subcanonicity: $s[t] \equiv \sigma(s[t]) \Rightarrow t \equiv \sigma(t)$
(v) variable conditions: $\sigma(x) \equiv x$, $\mathrm{Var}(u) \supseteq \mathrm{Var}(\sigma(u))$, and $\sigma(t\pi) \equiv \sigma(t)\pi$ for any renaming π of variables that preserves a fixed total ordering of the set of variables
A term t is called *canonical* if $t \equiv \sigma(t)$.

The ordering condition imposed in (v) is less restrictive than full invariance under variable renamings. For example, it still allows to canonize $y + x$ to $x + y$, provided that, say, $y \succ x$.

We now extend σ to a canonizer $\bar{\sigma}$ for terms over the combined signature. The usual proceeding is, given some Δ-top term $u[\![\vec{s}]\!]$, to: (i) recursively canonize $\vec{s}$, say such that $u[\![\bar{\sigma}(\vec{s})]\!] \equiv u[\![\vec{t}]\!]$, (ii) build an *abstraction* $u[\vec{x}]$ of $u[\![\vec{t}]\!]$ where the maximal non-theory terms are replaced by variables, (iii) canonize the theory term $u[\vec{x}]$, and (iv) reinsert the Φ-top terms t_i for the x_i in the result. In step (ii) the choice of variables may matter, because the canonizer need not be stable under arbitrary renamings of the variables. So an order-preserving bijection between terms and variables has to be given. We choose a simpler presentation: When dealing with theory terms, we directly consider Φ-top terms $f(\vec{u})$ as variables. Within this section, we name them *canonizer variables*, as opposed to *schematic variables*. The notation allows formulations like this:

Proposition 2. For theory terms s and t, that may contain canonizer variables, $s =_{\mathcal{T}} t$ implies $s =_{\mathcal{T}^{\Phi}} t$. Furthermore $s \neq_{\mathcal{T}} t$ entails $s \neq_{\mathcal{T}^{\Phi}} t$.

Now the extension of σ to terms over the combined signature can easily be expressed in an innermost fashion, as e. g. in [Kap02] or [SR02]:

Definition 3. $\bar{\sigma}(u[\![\vec{s}]\!]) :\equiv \sigma(u[\bar{\sigma}(\vec{s})])$, $\bar{\sigma}(f(\vec{u})) :\equiv f(\bar{\sigma}(\vec{u}))$, and $\bar{\sigma}(x) :\equiv \sigma(x)$.

Consider as an example the theory of linear rational arithmetic. A simple canonizer for $\mathcal{T}$ is obtained by increasingly arranging addends according to some fixed order on all variables and combining like terms. To compute $\bar{\sigma}(f(c) + 3g(1) - 5f(c+0))$, the canonizer variables $f(c)$, $g(1)$ and $f(c+0)$ are extracted and recursively canonized to $f(c)$, $g(1)$ and $f(c)$, respectively. The final result is $\sigma(f(c) + 3g(1) - 5f(c)) \equiv 3g(1) - 4f(c)$ if, say, $f(c)$ is greater than $g(1)$.

Evidently $\bar{\sigma}$ on theory terms is just σ. Soundness of the extension is easy to see. But completeness need not hold for non-convex theories, as exemplified by $\mathcal{T} = \{eq/2, T/0, F/0\}$ with $\sigma(eq(s,t)) \equiv T$ if $\sigma(s) \equiv \sigma(t)$, $\sigma(eq(s,t)) \equiv F$ otherwise and $\sigma(t) \equiv t$ in the remaining cases. With $\Phi = \{f/1\}$ we then have $f(eq(fT, fF)) =_{\mathcal{T}^{\Phi}} fF$ although left- and right-hand side are canonical. Very recently [KC03] have shown that the incompleteness vanishes if convexity is given:

Theorem 4. The extension of σ to $\bar{\sigma}$ preserves soundness, completeness, and computability for convex $\mathcal{T}$.

3.2 The black-box path ordering

In the sequel, we will need an ordering on mixed ground terms that comes as close as possible to a ground total reduction ordering, is compatible with canonizer applications, and has the so-called multiset property. We will define such an ordering in three steps: First, we define an auxiliary signature Φ^{κ}, whose symbols are, one the one hand, the free symbols in Φ, and on the other hand, the contexts over Δ. Then we define a variant of the lexicographic path ordering for terms over Φ^{κ} and prove its essential properties. Finally, we define a mapping from terms over $\Phi \cup \Delta$ to terms over Φ^{κ} and obtain an ordering for terms over $\Phi \cup \Delta$ via this mapping.

We assume a fixed set of canonizer variables $CV_{\text{norm}} = \{ c_i \mid i \in \mathbb{N} \}$ with the ordering $c_0 \prec_{\text{norm}} c_1 \prec_{\text{norm}} \ldots$. A ground term u over theory symbols and CV_{norm} is CV_{norm}-normalized, if the set of canonizer variables in u equals $\{ c_i \mid i < k \}$ for some $k \in \mathbb{N}$. Let Φ^{κ} be the signature that contains all function symbols from Φ and a new symbol κ_u for every CV_{norm}-normalized term u. Function symbols from Φ keep their arities; the arity of any κ_u is the number of occurrences of canonizer variables in u.

A well-founded total ordering $\succ_{\text{cont}}$ on Φ^{κ} is called a σ-context precedence, if it satisfies the following conditions for all function symbols $f \in \Phi$ and CV_{norm}-normalized terms u and v:

(a) $f \succ_{\text{cont}} \kappa_u$,

(b) if u and v contain identical multisets of canonizer variables, and if the number of non-canonical subterms in u is greater than the number of non-canonical subterms in v, then $\kappa_u \succ_{\text{cont}} \kappa_v$,
(c) if v is a proper subterm of u, then $\kappa_u \succ_{\text{cont}} \kappa_v$.

It is easy to construct a σ-context precedence for an arbitrary canonizer σ, for instance by comparing first the origin of the symbols (Φ vs. $\Phi^\kappa \setminus \Phi$), then, for κ_u, the number of non-canonical subterms in u, followed by the size of u, and finally by comparing the symbols according to some arbitrarily chosen total and well-founded ordering.

Let $\sim_\kappa$ be the congruence that is induced by the permutativity axioms $\kappa_u(x_1, \ldots, x_m) \simeq \kappa_u(x_{\pi(1)}, \ldots, x_{\pi(m)})$ for all κ_u.

The auxiliary ordering $\succ_\kappa$ on ground Φ^κ-terms induced by the σ-context precedence $\succ_{\text{cont}}$ is defined as follows: $s \succ_\kappa t$ if and only if

(a) $s \equiv h(s_1, \ldots, s_m)$, $h \in \Phi^\kappa$, and $s_i \succsim_\kappa t$ for some $i \in \{1, \ldots, m\}$, or
(b) $s \equiv f(s_1, \ldots, s_m)$, $f \in \Phi$, $t \equiv h(t_1, \ldots, t_n)$, $h \in \Phi^\kappa$ with $f \succ_{\text{cont}} h$ and $s \succ_\kappa t_j$ for all $j \in \{1, \ldots, n\}$, or
(c) $s \equiv f(s_1, \ldots, s_m)$, $t \equiv f(t_1, \ldots, t_m)$, $f \in \Phi$, $s \succ_\kappa t_j$ for all $j \in \{1, \ldots, n\}$, and there exists $i \in \{1, \ldots, m\}$ such that $s_1 \sim_\kappa t_1, \ldots, s_{i-1} \sim_\kappa t_{i-1}$ and $s_i \succ_\kappa t_i$, or
(d) $s \equiv \kappa_u(s_1, \ldots, s_m)$, $t \equiv \kappa_v(t_1, \ldots, t_n)$, and $\{s_1, \ldots, s_m\} \; (\succ_\kappa)^{\text{mul}} \; \{t_1, \ldots, t_n\}$, where $(\succ_\kappa)^{\text{mul}}$ is the multiset extension of $\succ_\kappa$ with respect to $\sim_\kappa$, or
(e) $s \equiv \kappa_u(s_1, \ldots, s_m)$, $t \equiv \kappa_v(t_1, \ldots, t_m)$, the multisets $\{s_1, \ldots, s_m\}$ and $\{t_1, \ldots, t_m\}$ agree up to $\sim_\kappa$, and $\kappa_u \succ_{\text{cont}} \kappa_v$.

The relation $\succ_\kappa$ differs from a traditional LPO with status only in that it compares two terms with top symbols in $\Phi^\kappa \setminus \Phi$ by comparing lexicographically first the multisets of arguments and then the two top symbols. One can show in essentially the same way as for the lexicographic path ordering (cf. [BN98, Chap. 5.4.2]) that $\succ_\kappa$ is a simplification ordering[1] on ground Φ^κ-terms which is compatible with $\sim_\kappa$ and total and irreflexive up to $\sim_\kappa$.

Let $\succ$ be an ordering on terms, and let $s \equiv u[\![s_1, \ldots, s_n]\!]$ be a term such that $\succ$ is total on the set $\{s_1, \ldots, s_n\}$. Then the CV_{norm}-normalized form of s is the term $\text{norm}(s, \succ)$ that is obtained from s by replacing the smallest term of $\{s_1, \ldots, s_n\}$ by c_0, the second smallest by c_1, and so on. (Multiple occurrences of the same term are replaced by the same canonizer variable from CV_{norm}.)

We define a binary relation $\succ_{\text{bb}}$ on ground terms over $\Phi \cup \Delta$ and a mapping κ from ground terms over $\Phi \cup \Delta$ to ground terms over Φ^κ by mutual recursion:

(a) $s \succ_{\text{bb}} t$ if and only if $\kappa(s)$ and $\kappa(t)$ are defined and $\kappa(s) \succ_\kappa \kappa(t)$.
(b) If $s \equiv f(s_1, \ldots, s_m)$, $f \in \Phi$, and $\kappa(s_i)$ is defined for all $i \in \{1, \ldots, m\}$, then $\kappa(s) \equiv f(\kappa(s_1), \ldots, \kappa(s_m))$.
(c) If $s \equiv u[\![s_1, \ldots, s_m]\!]$, $\kappa(s_i)$ is defined for all $i \in \{1, \ldots, m\}$, and $\succ_{\text{bb}}$ is total on the set $\{s_1, \ldots, s_m\}$, then $\kappa(s) \equiv \kappa_{\text{norm}(s, \succ_{\text{bb}})}(\kappa(s_1), \ldots, \kappa(s_m))$.

It is easy to check that κ is injective. Furthermore, if $t \equiv \kappa(s)$ for some term s, then the order of arguments of every κ_u occurring in t corresponds to the order

[1] In the sense of Middeldorp and Zantema [MZ94] for possibly infinite signatures.

of context variables in u. It is thus clear that the set of all terms that are the image of some term under κ contains at most one element of any $\sim_\kappa$-congruence class. This is the key argument in proving the following theorem:

Theorem 5. If $\succ_{\text{cont}}$ is a σ-context precedence, then the relation $\succ_{\text{bb}}$ and the mapping κ have the following properties for all ground Φ^κ-terms $r, s, t, t_1, \ldots, t_n$:

- (i) Transitivity: $r \succ_{\text{bb}} s$ and $s \succ_{\text{bb}} t$ imply $r \succ_{\text{bb}} t$
- (ii) Irreflexivity: $s \not\succ_{\text{bb}} s$.
- (iii) Well-foundedness: There exists no infinite decreasing $\succ_{\text{bb}}$-chain.
- (iv) Totality of $\succ_{\text{bb}}$: $s \succ_{\text{bb}} t$ or $s \prec_{\text{bb}} t$ or $s \equiv t$.
- (v) Totality of κ: $\kappa(s)$ and $\kappa(t)$ are defined.
- (vi) Subterm property: If t is a proper subterm of s, then $s \succ_{\text{bb}} t$.
- (vii) Compatibility with free contexts: $s \succ_{\text{bb}} t$ implies $f(\ldots, s, \ldots) \succ_{\text{bb}} f(\ldots, t, \ldots)$ for every $f \in \Phi$.
- (viii) Partial compatibility with theory contexts: $s \succ_{\text{bb}} t$ implies $u[\![\ldots, s \ldots]\!] \succ_{\text{bb}} u[\ldots, t, \ldots]$ if s is a Φ-top term.
- (ix) Multiset property: If s, $t_1, \ldots, t_n$ are Φ-top terms, $s \succ_{\text{bb}} t_i$ for every $i \in \{1, \ldots, m\}$, and u is a theory context, then $s \succ_{\text{bb}} u[\![t_1, \ldots, t_n]\!]$.

In order to ensure compatibility with canonizer applications, we have to impose an additional condition:

Assumption 6. From now on we require that the canonizer σ is *multiset decreasing* in the sense that $\sigma(u[\![s_1, \ldots, s_m]\!]) \equiv v[\![t_1, \ldots, t_n]\!]$ implies $\{s_1, \ldots, s_m\}$ $(\succeq_{\text{bb}})^{\text{mul}}$ $\{t_1, \ldots, t_n\}$, and $\sigma(u[\![s_1, \ldots, s_m]\!]) \equiv t$ implies $\{s_1, \ldots, s_m\}$ $(\succeq_{\text{bb}})^{\text{mul}}$ $\{t\}$ if t is a Φ-top term.

At first glance, this assumption looks a bit restrictive, in that it excludes canonizers which duplicate variables; e.g. when applying distributivity to turn $x(y+z)$ into $xy+xz$. However, the canonizer is free to internally employ a fresh theory operator $f_{\lambda uvw.uv+uw}$ with the semantics $f_{\lambda uvw.uv+uw}(x, y, z) =_{\mathcal{T}} xy + xz$ to resolve this non-linearity. This is possible, because in our calculus, theory operators must be known only to the ordering, the canonizer, and the solver that will be introduced in Sect. 3.4. For the ordering, dealing with infinite signatures is unproblematic, and canonizer and solver can always expand the new operator on the fly whenever they encounter it, continue as usual, and finally linearize again. For the remainder of the superposition calculus, the new theory operator is practically irrelevant: it treats theory contexts as black boxes anyway.

By Thm. 5, multiset decreasingness implies that $s[u[\![\vec{t}]\!]] \succeq_{\text{bb}} s[\sigma(u[\![\vec{t}]\!])]$ holds. Since $\bar{\sigma}(t)$ is obtained from t by iterated application of σ to Δ-top subterms, the following theorem is an obvious consequence:

Theorem 7. If σ is multiset decreasing, then $s[t] \succeq_{\text{bb}} s[\bar{\sigma}(t)]$.

Multiset decreasingness is crucial here. For canonizers σ that are not multiset decreasing, there need not exist any total ordering on mixed terms that is both compatible with canonizer applications and (partially) compatible with theory contexts. As an example consider $\Delta = \{h/2\}$, $\Phi = \{a/0, b/0\}$, and a canonizer that transforms any Δ-term t into $h(s, s)$, where s is the greatest canonizer variable in t. If $\succ$ is (partially) compatible with theory contexts and, say, $a \succ b$, then $h(a, a) \succ h(a, b)$, but on the other hand, σ transforms $h(a, b)$ back into $h(a, a)$.

Similarly, it is in general not possible to extend property (viii) of Thm. 5 to full compatibility with theory contexts while keeping totality and compatibility with canonizer applications. As an example, consider $\Delta = \{a/0, b/0, f/1, g/1\}$ and a canonizer σ that transforms $f(a)$ into $f(b)$ and $g(b)$ into $g(a)$. If $\succ$ were total and fully compatible with theory contexts, then either $a \succ b$ or $b \succ a$, but $a \succ b$ would imply $g(a) \succ g(b)$ and similarly $b \succ a$ would imply $f(b) \succ f(a)$.

In the following $\succ$ will denote the black-box path ordering $\succ_{\mathrm{bb}}$ to some arbitrary σ-context precedence $\succ_{\mathrm{cont}}$. As a further preparatory step, the black-box path ordering is lifted to literals and clauses as usually.

Definition 8. We successively define orderings on literals and clauses as follows:

(i) The *complexity* of an equation $s \simeq t$ is defined as the multiset $\{s, t\}$, and the complexity of a disequation $s \not\simeq t$ is $\{s, s, t, t\}$.

(ii) The *ordering* $\succ_L$ *on literals* is defined by comparing the complexities of the literals in the multiset extension of $\succ$.

(iii) The *clause ordering* $\succ_C$ takes the multisets of the literal complexities and compares them in the two-fold multiset extension of $\succ$.

3.3 Canonizing and rewriting

By now we have extended the canonizer to terms over the combined signature; and with the black-box path ordering we have developed a tool for canonizer-related termination argumentations. In this subsection we will first of all rephrase the extended canonizer as a rewrite system, i.e. in terms of equational deduction, second employ the black-box path ordering to show the resulting system is convergent, and third study convergence of special extensions. The rewrite system will be used as an ubiquitous simplification device and therefore later become part of the model construction.

With an extended canonizer $\bar{\sigma}$, we associate $\widehat{\sigma} = \bar{\sigma} \setminus \mathrm{id}$ as a ground rewrite system. So the rewrite relation $\longrightarrow_{\widehat{\sigma}}$ is just the context closure of the canonization $t \mapsto \bar{\sigma}(t)$ of non-canonical terms, and it has a very simple structure: If $s \longrightarrow_{\widehat{\sigma}} t$ is a top-level reduction, then $\bar{\sigma}(s) \equiv t$ and $s \not\equiv t$ and t is irreducible. By soundness and completeness of the extended canonizer (Thm. 4) the equality $\longleftrightarrow^*_{\widehat{\sigma}}$ generated by the rewrite system coincides with $=_{\mathcal{T}^\Phi}$.

The extended canonizer $\bar{\sigma}$ actually decides the word problem for $\mathcal{T}^\Phi$, via computation and syntactical comparison of normal forms. So does the rewrite-based reformulation:

Proposition 9. The rewrite system $\widehat{\sigma}$ is terminating and confluent.

Termination holds because the rewrite steps can be embedded in a black-box path ordering (Thm. 7) which itself is terminating (Thm. 5). Hence confluence is equivalent to local confluence, which reduces to joinability of critical pairs. The latter can be derived from the completeness of $\bar{\sigma}$ (Thm. 4).

We now study convergence of $\widehat{\sigma}$ extended by a single rule $l \Rightarrow r$ where l is a Φ-top term. Roughly this is the shape of equations after application of the solver. Within the analysis of confluence, critical pairs from such rules $l \Rightarrow r$ into canonizer rules will occur. For arbitrary terms t, l and r, we define recursively the *parallel reduction* $t[l{\mapsto}r]$ as r if $t \equiv l$, as x if $t \equiv x \not\equiv l$, and as $h(\vec{s}[l{\mapsto}r])$ if $t \equiv h(\vec{s}) \not\equiv l$. The following, somewhat technical proposition states that under specific requirements canonization and parallel reduction commute.

Proposition 10. Canonization and parallel reduction are related as follows:
(i) For terms over Δ, $v[x{\mapsto}u] =_{\mathcal{T}} \sigma(v)[x{\mapsto}u]$.
(ii) For terms over Δ and Φ, $s[l{\mapsto}r] =_{\mathcal{T}^\Phi} \bar{\sigma}(s)[l{\mapsto}r]$ in case that l is a canonical Φ-top term, and the strict subterms of s are canonical.

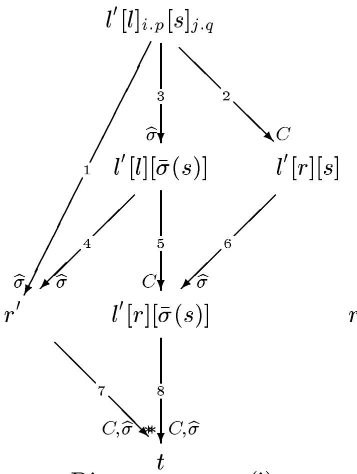
The first statement is a consequence of the soundness of σ. The second one relies on the subcanonicity of σ and $\bar{\sigma}$, and follows from the first one by Prop. 2. It is an important ingredient to the proof of the following confluence result. This theorem is at the heart of our approach; we therefore give a detailed proof.

Theorem 11. For every ground rewrite rule $l \Rightarrow r$ where l is a canonical Φ-top term and $l \succ r$, the combined system $\widehat{\sigma} \cup \{l \Rightarrow r\}$ is terminating and confluent.

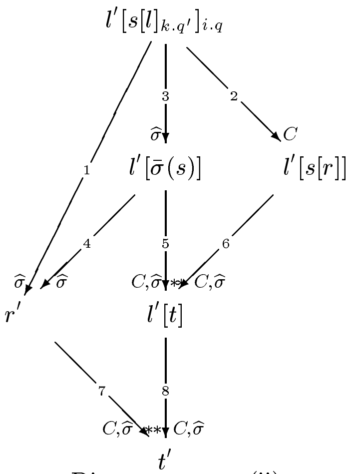
Proof. Because of Thm. 5 we have that $t[l] \succ t[r]$ for every term t; therefore $\widehat{\sigma} \cup \{l \Rightarrow r\}$ is terminating. Hence the system is confluent if all the critical pairs are joinable. We prove the latter by induction on the peak term wrt. $\succ$.

The critical pairs of $\widehat{\sigma}$ are joinable by Prop. 9. The rule $l \Rightarrow r$ is ground and overlaps with itself trivially only. So we only need to study critical pairs between $l \Rightarrow r$ and some $l' \Rightarrow r' \in \widehat{\sigma}$. For short $\longrightarrow_C := \longrightarrow_{\{l \Rightarrow r\}}$, and $\longrightarrow_{C,\widehat{\sigma}} := \longrightarrow_{\{l \Rightarrow r\} \cup \widehat{\sigma}}$.

Canonizer steps below rewrite steps like in $r\ {}_C\!\longleftarrow l[l'] \longrightarrow_{\widehat{\sigma}} l[r']$ are not possible because $l[l']$ is canonical by assumption and hence also the subterm l' is. Now let us analyze vice versa rewrite steps below canonizer steps, i. e. $r'\ {}_{\widehat{\sigma}}\!\longleftarrow l'[l]_{i.p} \longrightarrow_C l'[r]$. Since the subterms of l' need not be canonical, we distinguish the following cases:



Divergence case (i)



Divergence case (ii)

(i) l' might contain a $\longrightarrow_{\bar{\sigma}}$-redex at a position $j.q$ parallel to $i.p$. The divergency corresponds to the reduction steps numbered 1 and 2 in the diagram, whereas the $\longrightarrow_{\bar{\sigma}}$-redex leads to the one numbered 3. Step 4 exists because $l'[l][s]$, r' and $l'[l][\bar{\sigma}(s)]$ are $\mathcal{T}^{\Phi}$-equal and r' is canonical. Steps 5 and 6 are possible because reductions 2 and 3 have been parallel. Steps 7 and 8 must exist inductively because reduction 3 is decreasing.

(ii) Since l is canonical, l' has no $\longrightarrow_{\bar{\sigma}}$-redex below $i.p$, but there may exist one strictly above, say at $i.q$ where $p = q.k.q'$. This corresponds to reduction 3 in the diagram. As above, Step 4 exists because $l'[s[l]]$, r' and $l'[\bar{\sigma}(s[l])]$ are $\mathcal{T}^{\Phi}$-equal and r' is canonical. Steps 5 and 6 are justified inductively by $l' \succ_{\text{sub}} s$ and $\succ \supseteq \succ_{\text{sub}}$; and so are steps 7 and 8 because reduction 3 is decreasing.

$$\begin{tikzcd}
 & l'[l]_{i.p} \arrow[dl, "1", "\bar{\sigma}"'] \arrow[dr, "2"', "C"] & \\
r' \equiv \bar{\sigma}(l') \arrow[d, "3", "C*"'] & & l'[r] \arrow[d, "4"', "C*"] \\
\bar{\sigma}(l')[l\mapsto r] \arrow[dr, "5", "\bar{\sigma}*"'] & & l'[l\mapsto r] \arrow[dl, "6"', "*\bar{\sigma}"] \\
 & t &
\end{tikzcd}$$

Divergence case (iii)

(iii) Finally consider l' without any $\longrightarrow_{\bar{\sigma}}$-redex but on top-level. Left-hand and right-hand side of the divergency can sequentially be reduced by $\longrightarrow_{C}$ to their respective images under the parallel reduction $[l\mapsto r]$. By Prop. 10, these are $\mathcal{T}^{\Phi}$-equal and hence have the same canonical form. $\square$

3.4 The solver, and its extension and application

Besides the canonizer, the solver is the second procedural component of Shostak's method, basically a unitary unification algorithm. From a conceptual point of view, for Shostak's method itself the solver is even of higher importance, since there the canonizer can be seen as a facultative simplification device. The solver is characterized as follows (cf. [Gan02]):

Assumption 12. For any equation $s \simeq t$ over Δ, the solver returns

$$\mathsf{solve}(s \simeq t) = \begin{cases} \{\} & s =_{\mathcal{T}} t \\ \bot & s \neq_{\mathcal{T}} t \\ \{\vec{x} \simeq \vec{u}\} & \text{otherwise} \end{cases}$$

where in the last case the following conditions additionally hold:

(i) $\{\vec{x} \simeq \vec{u}\} \neq \{\}$ (ii) $x_i \not\equiv x_j$ for $i \neq j$

(iii) $\{\vec{x}\} \subseteq \mathrm{Var}(s \simeq t)$ (iv) $\{\vec{x}\} \cap \mathrm{Var}(\vec{u}) = \{\}$

(v) $\mathcal{T} \models \forall X.(s \simeq t \leftrightarrow \exists Y.\vec{x} \simeq \vec{u})$

with $Y = \mathrm{Var}(\vec{u}) \setminus \mathrm{Var}(\vec{s} \simeq \vec{t})$ and $X \cap Y = \{\}$

We want to employ this special unification algorithm to solve equations over the combined signature with respect to one (or more) of their theory subterms. To this end, we extend it like the canonizer by considering all maximal Φ-top terms as canonizer variables.

Definition 13. For equations $s \simeq t$ over the combined signature $\Delta \cup \Phi$, we define $\overline{\mathsf{solve}}(s \simeq t) := \mathsf{solve}(s \simeq t)$, where $s \simeq t$ on the right-hand side is read as an equation over Δ.

In our example of linear arithmetic, let us consider for instance a call of $\overline{\mathsf{solve}}$ on the equation $f(c) + 3g(1) - 5f(c+0) \simeq 0$. The canonizer variables in this equation are $f(c)$, $g(1)$ and $f(c+0)$. So the extended solver can for example return the equation $f(c+0) \simeq \frac{1}{5}f(c) + \frac{3}{5}g(1)$.

With arguments similar to those required in the proof of Prop. 2, we get the following property which shows that the extension serves the intended purpose:

Proposition 14. The extended solver is sound in the sense that
$\overline{\mathsf{solve}}(s \simeq t) = \{\}$ implies $s =_{\mathcal{T}^\Phi} t$,
$\overline{\mathsf{solve}}(s \simeq t) = \perp$ implies $s \neq_{\mathcal{T}^\Phi} t$, and
$\overline{\mathsf{solve}}(s \simeq t) = \{\vec{x} \simeq \vec{u}\}$ implies $\mathcal{T}^\Phi \models \forall X_0.(s \simeq t \leftrightarrow \exists Y. \vec{x} \simeq \vec{u})$,
where X_0 is the set of schematic variables of X of Ass. 12.

Now we study the application of the solver to equations in clausal contexts. The purpose of the solver is to extract maximal Φ-top subterms from theory contexts. Indeed, if we apply it to a ground equation and obtain a solution $\{\vec{x} \simeq \vec{u}\}$, then each of the x_i is actually a canonizer variable, i.e. a Φ-top term, whereas the variables within $\vec{u}$ are existentially quantified. We formalize this as follows:

Definition 15. Consider a ground clause $C = D \vee s \simeq t$. Depending on the outcome of $\overline{\mathsf{solve}}(s \simeq t)$, the solved form of C at $s \simeq t$ is defined as:
(i) if the result is $\{\}$, the tautological clause $\top$;
(ii) in case of $\perp$, the subclause D;
(iii) for $\{\vec{x} \simeq \vec{u}\}$, each of the clauses $D \vee x_i \simeq \widehat{u}_i$ where $\widehat{\vec{u}}$ denotes $\vec{u}$ after consistently replacing existentially quantified variables with fresh Skolem functions.

Note that because of the restriction to ground clauses, only Skolem constants will occur in (iii); they are shared between different solved forms only if they arise from the same Skolemization. We consider them as part of the free signature Φ. The effect of the solving operation on semantics still needs to be characterized. Proposition 14 and the fact that the interpretation of the Skolem constants can be chosen suitably imply that a clause is satisfiable if and only if the set of its solved form is:

Proposition 16. If an interpretation $I^\Phi \in \mathcal{T}^\Phi$ satisfies $D \vee s \simeq t$, then there is an extension of I^Φ to the Skolem constants occurring in any of the solved forms such that the extension satisfies every solved form. Conversely, if there is such an extension of I^Φ, then I^Φ satisfies $D \vee s \simeq t$.

The solving operation still has to be related to the ordering structure that is imposed on terms by the black-box path ordering: First, we want to ensure that solving and canonizing a clause can be done in finite time, i.e., solving should be non-increasing. Second, the outcome $x_i \simeq \widehat{u}_i$ shall be interpreted as a form of definition of x_i and applied in left-to-right direction; and in order to achieve this, the definition must be decreasing.

Assumption 17. We require that whenever $D \vee x_i \simeq \widehat{u}_i$ is a solved form of $D \vee s \simeq t$, then $s \simeq t \succeq_L x_i \simeq \widehat{u}_i$ and $x_i \succ \widehat{u}_i$ hold.

The restriction is always met if the complex solutions $\{\vec{x} \simeq \vec{u}\}$ that $\overline{\mathsf{solve}}(s \simeq t)$ returns are unitary and x is the maximal canonizer variable of the equation, as is for instance the case for linear arithmetic: We get $x \succ \widehat{u}$ by the multiset property of the black-box path ordering $\succ$, and $s \simeq t \succeq_L x \simeq \widehat{u}$ additionally by its subterm property.

Finally we now stipulate how to apply the solver to literals in clausal contexts: eagerly, but to strictly maximal equations only.

Convention 18. Every clause $C \equiv D \vee s \simeq t$ where $s \simeq t$ is strictly maximal, is without loss of generality presented such that
(i) C equals its solved form at $s \simeq t$, and
(ii) s is canonical.

Indeed every clause C which does not adhere to the conditions can be transformed into an equisatisfiable set of clauses that does: Until the condition is met, C is repeatedly replaced by the canonized set of all the corresponding solved forms, which by Ass. 12 is always finite. By Ass. 17 and König's Lemma this replacing process terminates; and Prop. 16 guarantees equisatisfiability. The unit clause $f(c) + 3g(1) - 5f(c+0) \simeq 0$ of linear arithmetic is first solved to $f(c+0) \simeq \frac{1}{5}f(c) + \frac{3}{5}g(1)$, then canonized to $f(c) \simeq \frac{1}{5}f(c) + \frac{3}{5}g(1)$ and finally solved to $f(c) \simeq \frac{3}{4}g(1)$. For a clause D that explicitly needs to be reshaped, as is the case for conclusions in inference rules, by $\Gamma(D)$ we denote the application of a such a reshaping postprocessor. Since a clause can have more than one solved form, we may view Γ as a non-deterministic function returning one of them.

4 Superposition

4.1 Ground case

In the following we restrict our attention to ground clauses. With the machinery developed so far, extending the superposition calculus to work modulo a Shostak theory $\mathcal{T}^{\Phi}$ is more or less straightforward, in that the argumentations for the standard calculus smoothly carry over. Mainly we have to integrate the canonizer into the model construction process and ensure that the coherence pairs are trivial. This will be possible because maximal equations within clauses are solved such that they have a Φ-top left-hand side. As was shown in Sect. 3.3, extending $\widehat{\sigma}$ by such rules preserves confluence. The presentation of our calculus follows the lines of [NR01].

Definition 19. The *superposition calculus modulo* $\mathcal{T}^{\Phi}$ for ground clauses consists of the following inference rules:
(i) *Superposition:*

$$\frac{C \vee l \simeq r \quad D \vee s[l] \bowtie t}{\Gamma(C \vee D \vee s[r] \bowtie t)}$$

if l and $l \simeq r$ are strictly maximal in the first premise, and s and $s \bowtie t$ are maximal in the second and strictly maximal for $\bowtie = \simeq$

(ii) *Equality resolution:*

$$\frac{C \vee s \not\simeq s}{\Gamma(C)} \quad \text{if } s \not\simeq s \text{ is maximal}$$

(iii) *Equality factoring:*

$$\frac{C \vee s \simeq t \vee s \simeq t'}{\Gamma(C \vee t \not\simeq t' \vee s \simeq t')} \quad \text{if } s \simeq t \text{ is maximal, and } s \succ t$$

To get a flavour of how derivations in this calculus proceed, consider the clause set $\{f(a-1)-1 \simeq a+1; f(b)-1 \not\simeq a+1 \vee f(b) \simeq b-2; b+1 \simeq a\}$. Assume $f(b) \succ a \succ b$. Reshaping the clauses according to our convention we obtain (1) $\underline{f(a-1)} \simeq a+2$, (2) $\underline{f(b)} \not\simeq a+2 \vee f(b) \simeq b-2$ and (3) $\underline{a} \simeq b+1$, where maximal terms in maximal literals are underlined. From (1) and (3) we obtain by the superposition rule (4) $\underline{f(b)} \simeq a+2$, which with (2) gives $a+2 \not\simeq a+2 \vee \underline{f(b)} \simeq b-2$. This clause and (4) lead to $\underline{a+2} \not\simeq \underline{a+2} \vee a+2 \simeq b-2$. From this, equality factoring deduces $\underline{a} \simeq b-4$. With (3) we obtain the empty clause.

We now employ the proof technique of *model generation* of [BG94] to show that every saturated clause set free of the empty clause has a model. This model will be a congruence induced by a convergent ground rewrite system. The canonizer $\widehat{\sigma}$ is part of that system. In the sequel S denotes an arbitrary, fixed set of ground clauses according to Conv. 18. Every contained clause may have to contribute a rule to become valid in the model:

Definition 20. For $C = C' \vee l \simeq r \in S$, we define $\mathrm{Gen}(C) = \{l \Rightarrow r\}$ if the following conditions hold, and $\mathrm{Gen}(C) = \{\}$ otherwise:
(i) $(R_C \cup \widehat{\sigma})^* \not\models C$,
(ii) $l \succ r$, and $l \simeq r$ is strictly maximal in C,
(iii) l is irreducible with respect to R_C,
(iv) $R^*_{C,\widehat{\sigma}} \not\models t \simeq r$ for all $l \simeq t$ in C;
where $R_C = \bigcup_{D \prec_C C} \mathrm{Gen}(D)$, and $(\_)^*$ stands for closure under congruence axioms. If C has no positive literal, let $\mathrm{Gen}(C) = \{\}$. Then $R_S := \bigcup_{D \in S} \mathrm{Gen}(D)$. For short $R_{S,\widehat{\sigma}} := R_S \cup \widehat{\sigma}$, and $R_{C,\widehat{\sigma}} := R_C \cup \widehat{\sigma}$.

Now the shape of a generated rule $l \Rightarrow r$ is the following: By Conv. 18, the containing clause is solved at $l \simeq r$. By construction of $\overline{\mathsf{solve}}$, it is not possible that both l and r are ground theory terms. Because of $l \succ r$ and Ass. 17, the left-hand side l must be a Φ-top term. We summarize:

Proposition 21. If $l \Rightarrow r$ is a generated rule, then l is a canonical Φ-top term.

The next object of study is $R_{S,\widehat{\sigma}}$ which comprises the canonizer-related rules of $\widehat{\sigma}$ and the generated rules. By construction it is embedded in the black-box path ordering $\succ$ and therefore terminating. This allows to deduce confluence from the joinability of critical pairs. Divergencies between rules of $\widehat{\sigma}$ converge due to Prop. 9. Pairs between $\widehat{\sigma}$ and generated rules are joinable by Thm. 11; in that sense, R_S and $\widehat{\sigma}$ are coherent. R_S is confluent by the same argumentation as for standard superposition.

Lemma 22. The rewrite system $R_{S,\widehat{\sigma}}$ is terminating and confluent.

Finally, the congruence $R^*_{S,\widehat{\sigma}}$ generated by this rewrite system exhibits a $\mathcal{T}^\Phi$-model for S under appropriate conditions: namely S being closed under superposition and not containing the empty clause. By Prop. 21 the restriction of $R^*_{S,\widehat{\sigma}}$ to Δ is just $\widehat{\sigma}^*$, which is in $\mathcal{T}$ because $\mathcal{T}$ contains its σ-models. Similar to the proof in [NR01, p. 388f] one can show that $R^*_{S,\widehat{\sigma}} \models C$ for every clause $C \in S$, essentially via induction on $\succ_C$ and case distinction as to the occurrence of the maximal literal. Additionally one has to take into account that the postprocessor turns the syntactic conclusion into an equisatisfiable set of solved forms which are smaller or equal. Summing it up, unsatisfiable clause sets are just those that cannot be closed under superposition without producing the empty clause. Hence we have in the usual sense:

Theorem 23. Superposition modulo $\mathcal{T}^\Phi$ is refutationally complete for ground clauses with respect to $\mathcal{T}^\Phi$.

But recall that this requires that the ordering restriction imposed in Ass. 17 has been met for any call of the solver. For situations in which this cannot be achieved, a possible way of circumventing the problem may consist in considering some of the coherence pairs.

Let us briefly remark that the concept of *redundancy* smoothly carries over from the ordinary superposition calculus, as well as the concept of *selection*. Just like the standard inference system, superposition modulo $\mathcal{T}^\Phi$ is turned into a decision procedure for $\mathcal{T}^\Phi$-satisfiability of ground Horn clauses by means of eager selection (cf. [NR01, p. 411f]). Via splitting of clauses, this can be extended to a decision procedure for arbitrary ground clauses.

4.2 Towards non-ground clauses

The calculus that we have considered so far works on ground clauses. In non-ground clauses, schematic variables are universally quantified within their clauses, and each clause represents the set of all its canonized ground instances. If we want to extend our calculus to non-ground clauses, then inferences between these clauses must represent all necessary inferences between their instances in a finite way. In the usual superposition calculus, this happens essentially by replacing equality tests in the inference rules by unification. In our case, it is also necessary to extend the other basic operations of the calculus to non-ground terms or formulas.

Extending the black-box path ordering to non-ground terms is somewhat tedious but possible in principle. Of course, the extended ordering is not total on non-ground terms, so that maximal terms or literals in a clause are no longer uniquely determined. One has to beware of the following pitfall, though: If a schematic variable occurs directly below a theory context, then in a ground instance, the variable may be replaced by a Δ-top term, thus enlarging the theory context. Such a context is therefore smaller than free function symbols, but essentially incomparable with other contexts (unless required by the subterm property).

Can we also extend the canonizer and the solver to non-ground terms and literals? If schematic variables occur only directly below free function symbols, the answer is again positive. If the ordering is not total on the subterms of a theory context, a case split on the ordering relations possible may lead to more than one result, but the number of resulting terms or formulas is always finite.[2] On the other hand, schematic variables occurring directly below theory contexts will now lead to failure: in a ground instance, they may be replaced by Δ-top terms, thus enlarging the theory context, and the set of all canonizations or solved forms of these ground instances is in general not finitely representable. This case can only be handled if more information on the internals of the canonizer and the solver is available. Note however that canonization and solving are relevant only for maximal or selected literals; so schematic variables in other literals are always unproblematic.

5 Conclusion

In this paper, we have integrated a Shostak theory into the superposition calculus. The Shostak-style components for deciding the clausal validity problem, i. e. canonizer and solver, have been employed as simplification devices, so that no coherence pairs between theory axioms and other axioms have to be considered. Under the assumption that the solver meets some ordering restriction, as is e. g. the case for linear arithmetic, our calculus is refutationally complete on mixed ground clauses. Ongoing work is to liberalize these restrictions and to develop practically useful solutions to the lifting problem.

References

[BG94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BGW94] L. Bachmair, H. Ganzinger, and U. Waldmann. Refutational theorem proving for hierarchic first-order theories. *Applicable Algebra in Engineering, Communication and Computing*, 5(3/4):193–212, 1994.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[Bür90] H.-J. Bürckert. A resolution principle for clauses with constraints. In M. E. Stickel, editor, *Proceedings of the 10th International Conference on Automated Deduction*, volume 449 of *LNAI*, pages 178–192. Springer-Verlag, 1990.

[Gan02] H. Ganzinger. Shostak light. In A. Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction*, volume 2392 of *LNAI*, pages 332–346. Springer-Verlag, 2002.

[2] To improve the efficiency, inheriting ordering constraints may be useful.

[GHW03] H. Ganzinger, Th. Hillenbrand, and U. Waldmann. Superposition modulo a Shostak theory. Technical report, Max-Planck-Institut für Informatik, Saarbrücken, 2003.

[GN00] G. Godoy and R. Nieuwenhuis. Paramodulation with built-in abelian groups. In *Proceedings of the 15th IEEE Symposium on Logic in Computer Science*, pages 413–424. IEEE Computer Society Press, 2000.

[JM92] J.-P. Jouannaud and C. Marché. Termination and completion modulo associativity, commutativity and identity. *Theoretical Computer Science*, 104(1):29–51, 1992.

[Kap02] D. Kapur. A rewrite rule based framework for combining decision procedures. In A. Armando, editor, *Proceedings of the 4th International Workshop on Frontiers of Combining Systems*, volume 2309 of *LNAI*, pages 87–102. Springer-Verlag, 2002.

[KC03] S. Krstić and S. Conchon. Canonization for disjoint unions of theories. In *Proceedings of the 19th International Conference on Automated Deduction*, LNAI. Springer-Verlag, 2003. This volume.

[LC86] Ph. Le Chenadec. *Canonical Forms in Finitely Presented Algebras.* John Wiley & Sons, 1986.

[Mar94] C. Marché. Normalised rewriting and normalised completion. In *Proceedings of the 9th IEEE Symposium on Logic in Computer Science*, pages 394–403. IEEE Computer Society Press, 1994.

[MZ94] A. Middeldorp and H. Zantema. Simple termination revisited. In *Proceedings of the 12th International Conference on Automated Deduction*, volume 814 of *LNAI*, pages 451–465. Springer-Verlag, 1994.

[NO79] G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979.

[NR01] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 7, pages 371–443. Elsevier, 2001.

[Plo72] G. D. Plotkin. Building-in equational theories. In B. Meltzer and D. Michie, editors, *Machine Intelligence 7*, chapter 4, pages 73–90. American Elsevier, 1972.

[RS01] H. Rueß and N. Shankar. Deconstructing Shostak. In *Proceedings of the 16th IEEE Symposium on Logic in Computer Science*, pages 19–28. IEEE Computer Society Press, 2001.

[Sho84] R. E. Shostak. Deciding combinations of theories. *Journal of the ACM*, 31(1):1–12, 1984.

[SR02] N. Shankar and H. Rueß. Combining Shostak theories. In S. Tison, editor, *Proceedings of the 13th International Conference on Rewriting Techniques and Applications*, volume 2378 of *LNCS*, pages 1–18. Springer-Verlag, 2002.

[Sti85] M. E. Stickel. Automated deduction by theory resolution. *Journal of Automated Reasoning*, 1(4):333–355, 1985.

[Stu00] J. Stuber. Deriving theory superposition calculi from convergent term rewriting systems. In L. Bachmair, editor, *Proceedings of the 11th International Conference on Rewriting Techniques and Applications*, volume 1833 of *LNCS*, pages 229–245. Springer-Verlag, 2000.

[Wal02] U. Waldmann. Cancellative abelian monoids and related structures in refutational theorem proving (Part I). *Journal of Symbolic Computation*, 33(6):777–829, 2002.