Modular Termination and Combinability for Superposition Modulo Counter Arithmetic

Christophe Ringeissen, Valerio Senni*

LORIA-INRIA Nancy Grand Est
E-mail: FirstName.LastName@loria.fr

Abstract. Modularity is a highly desirable property in the development of satisfiability procedures. In this paper we are interested in using a dedicated superposition calculus to develop satisfiability procedures for (unions of) theories sharing counter arithmetic. In the first place, we are concerned with the termination of this calculus for theories representing data structures and their extensions. To this purpose, we prove a modularity result for termination which allows us to use our superposition calculus as a satisfiability procedure for combinations of data structures. In addition, we present a general combinability result that permits us to use our satisfiability procedures into a non-disjoint combination method à la Nelson-Oppen without loss of completeness. This latter result is useful whenever data structures are combined with theories for which superposition is not applicable, like theories of arithmetic.

1 Introduction

Software verification tasks require the availability of solvers that are able to discharge proof obligations involving data-structures together with arithmetic constraints and other mathematical abstractions, such as size abstractions. Besides, the use of Satisfiability Modulo Theories (SMT) solvers allows us to focus on the development of satisfiability procedures for such mixed theories. In this setting, the problem of designing the satisfiability procedures is often addressed with success by using approaches based on combination [15].

Problems arise when we consider combinations involving theories whose signatures are non-disjoint. This is especially the case when we consider theories sharing some algebraic constraints [14,16,17,18,20,21]. In order to combine satisfiability procedures for the single theories to handle constraints in their non-disjoint union one needs to rely on powerful methods such as the combination framework of [9,10]. These methods are based on semantic properties of the considered theories, such as compatibility and computability of bases of the shared entailed equalities, which often require complex proofs.

A further issue concerns the development of correct and efficient satisfiability procedures for the single theories, possibly using a systematic approach. In this regard, the use of superposition calculus has proved to be effective

* The author acknowledges support from ERCIM during his stay at LORIA-INRIA.

to deal with classical data structures, which do not involve arithmetic constraints [1,2,4,5,6,13].

In this paper we address both aspects by: (1) considering a superposition calculus with a built-in theory of counter arithmetic [17,18] and (2) providing modularity results for termination and combinability, based on conditions on the saturations of the component theories that can be checked automatically.

Our contributions are twofold. First, we prove a modular termination result for extending the applicability of the superposition calculus to theories that share a theory of counter arithmetic. This generalizes, to the non-disjoint case, the results in [1], where the authors consider the standard superposition calculus and signature-disjoint theories. This result allows us to drop some of the complex conditions required by the combination framework when we deal with theories that can be treated uniformly through superposition.

Second, we prove a general compatibility result that allows us to use our superposition-based satisfiability procedures into the combination framework of [10]. We prove that any satisfiability procedure obtained by using our modular termination result is able to compute a finite basis of the shared entailed equalities. In addition, we provide a sufficient condition on the form of the saturations of the theories that allows us to conclude compatibility of the component theories with respect to the shared theory and, thus, completeness of their combination.

As an outcome, we have less and simpler restrictions on combinability and we are able to obtain satisfiability procedures both by a uniform approach for theories that can be treated well by superposition (e.g., data structures) and by combination with other solvers for theories which are not 'superposition-friendly' (such as theories of arithmetic).

To show the application of our results in practice, we introduce a class of new theories modeling data structures and equipped with a counting operator that allows us to keep track of the number of the modifications (writes, constructors, etc.) performed on a data structure. In these theories we are able to distinguish between versions of the same data structure obtained by some update.

The paper is organized as follows. In Section 2 we briefly introduce an example in which we use data structures equipped with a mechanism to count the update operations. In Section 3 we introduce some basic notions and recall the superposition calculus for counter arithmetic. In Section 4 we present our modular termination result. In Section 5 we present our general compatibility result. In Section 6 we discuss in details how these results can be applied to our motivating example. Section 7 concludes with some perspectives.

2 A Motivating Example

Let us now consider an example where we show the application of our technique to the analysis of a function *minmax*, defined as follows:

```
function minmax (l : LIST) : RECORD {
  while (l != nil) {
```

```
        e := car(l);
        if e < rselect1(r) then r := rstore1(r, e);
        if rselect2(r) < e then r := rstore2(r, e);
        l := cdr(l)
      };
      return r
  }
```

The function *minmax* stores into a binary record the maximum and minimum elements of a given list of rational numbers. We consider a theory of lists T_{LV} (including the classic car and cons operators) and a theory of records T_{RV} (including the the classic $\mathsf{rselect}_\mathsf{i}$ and $\mathsf{rstore}_\mathsf{i}$ operators), both equipped with a counting operator: $\mathsf{count}_R(r)$, which denotes the number of updates performed on the record r, and $\mathsf{count}_L(l)$, which denotes the number of elements inserted into the list l and coincides with the size of the list. In order to verify the correctness of the *minmax* function, we will prove that $\psi\colon\ \forall\, l, r\, (r = minmax(l) \Rightarrow \mathsf{count}_R(r) \leq \mathsf{count}_L(l))$ holds. The meaning of ψ is that the record r will not be updated more than 'size of l' times.

We will prove the desired property by relying on an SMT solver modulo $(T_{LV} \cup T_{RV} \cup T_S) \cup T_{\mathbb{Q}}$, where T_S is a (shared) theory of counter arithmetic and $T_{\mathbb{Q}}$ is a theory extension of T_S corresponding to the theory of linear arithmetic over the rationals. We develop a satisfiability procedure for $(T_{LV} \cup T_{RV} \cup T_S) \cup T_{\mathbb{Q}}$ in two steps. In the first step we use our result on modular termination to obtain a superposition-based satisfiability procedure for $T_{LV} \cup T_{RV} \cup T_S$. Then, in the second step, we use our general compatibility result to show that the requirements needed to combine $T_{LV} \cup T_{RV} \cup T_S$ with $T_{\mathbb{Q}}$ by using the non-disjoint combination framework of [10] are fulfilled. In Section 6 we discuss these aspects in detail and we also show that, in order to prove ψ we need to add some extra assumptions, namely that r is a 'fresh' record (no update operations have been performed on it) and r has been initialized so that $\mathsf{rselect}_1(r) \leq \mathsf{rselect}_2(r)$.

3 Preliminaries

Let us consider a many-sorted language. A *signature* Σ is a set of sorts, function and predicate symbols (each endowed with the corresponding arity and sort). We assume that, for each sort s, the equality '$\simeq_s$' is a logical constant that does not occur in Σ and that is always interpreted as the identity relation over (the interpretation of) s; moreover, as a notational convention, we will often omit the subscript for sorts and we will use the symbol $\bowtie$ to denote either $\simeq$ or $\not\simeq$. The signature obtained from Σ by adding a set $\underline{a}$ of new constants (i.e., 0-ary function symbols, each of them equipped with its sort) is denoted by $\Sigma^{\underline{a}}$ and named a *constant expansion* of Σ. *Σ-atoms*, *Σ-literals*, *Σ-clauses*, and *Σ-formulae* are defined in the usual way (see, e.g., [8]). The empty clause is denoted by $\bot$. A set of Σ-literals is called a *Σ-constraint*. Terms, literals, clauses and formulae are said to be *ground* whenever no variable appears in them; *sentences* are formulae in which free variables do not occur. Given a function

symbol f, a f-rooted term is a term whose top-symbol is f. Given a term t and a position p, $t_{|p}$ denotes the subterm of t at position p, and $t[l]_p$ denotes the term t in which l appears as the subterm at position p. The *depth* of a term t is defined as follows: $depth(t) = 0$, if t is a constant or a variable, and $depth(f(t_1, \ldots, t_n)) = 1 + \max\{depth(t_i) \mid 1 \leq i \leq n\}$. The *depth* of a literal $l \bowtie r$ is $depth(l \bowtie r) = depth(l) + depth(r)$.

In order to define models, we rely on the standard notion of a *Σ-structure* $\mathcal{M}$, which consists of: (1) a typed domain D, that is a domain partitioned into a (finite) set of (sub)domains, one for each sort, and (2) a sort- and arity-matching interpretation $\mathcal{I}$ of the function and predicate symbols from Σ. The truth of a Σ-formula in the structure $\mathcal{M}$ is defined in any of the standard ways.

A Σ-theory T is a collection of Σ-sentences, called the axioms of T. If every axiom is a sentence of the form $\forall \overline{x}\, A$, where A is a quantifier free formula, then we say that the theory is *universal*. An *equational theory* is a universal theory whose axioms are universally quantified equalities.

In this paper, we are concerned with the (*constraint*) *satisfiability problem* for a given theory T, also called the T-satisfiability problem, which is the problem of deciding whether a Σ-constraint is satisfiable in a model of T (and, if so, we say that the constraint is T-satisfiable). Note that a constraint may contain variables: since these can be replaced by fresh new constants (preserving satisfiability), we can reformulate the constraint satisfiability problem as the problem of deciding whether a finite conjunction of ground literals in a constant expansion Σ^a is true in a Σ^a-structure whose Σ-reduct is a model of T.

We consider inference systems using well-founded orderings on terms and literals. An ordering $\succ$ on terms is a *simplification ordering* [7] if it is stable ($l \prec r$ implies $l\sigma \prec r\sigma$ for every substitution σ), monotonic ($l \prec r$ implies $t[l]_p \prec t[r]_p$ for every term t and position p), and has the subterm property (i.e., it contains the subterm ordering: if l is a strict subterm of r, then $l \prec r$). Simplification orderings are well-founded. A term t is *maximal* in a multiset S of terms if $t \not\preceq u$, for every $u \in S$ different from t. An ordering on terms is extended to literals by using its multiset extension on literals viewed as multisets of terms. Any positive literal $l \simeq r$ (resp. negative literal $l \not\simeq r$) is viewed as the multiset $\{l, r\}$ (resp. $\{l, l, r, r\}$). Also maximality is extended to literals, by defining a term l maximal in a literal whenever l is maximal in the corresponding multiset.

3.1 Superposition Calculus for Counter Arithmetic

Recent literature has focused on the use of superposition calculus to decide the satisfiability of ground formulae in theories extending the theory of Integer Offsets [1,4]. These techniques are based on a problem-specific reduction of the input set of clauses to a new (equisatisfiable) one that admits a finite axiomatization of the successor function. Then, the standard superposition calculus [3] can be used as a decision procedure for the satisfiability of the obtained set of clauses. Moreover, these approaches allow the combination with other superposition-based decision procedures and ensure termination whenever the involved theories satisfy the so-called 'variable inactivity' property and are signature-disjoint.

In contrast, we are interested in a superposition-based calculus that is able to cope with *non-disjoint* extensions of a theory of Counter Arithmetic.

Theories of Counter Arithmetic. T_S is the theory of Increment, which defines the behavior of the successor function s and the constant 0. T_S has the monosorted signature $\Sigma_S := \{0 : \text{NUM}, \mathsf{s} : \text{NUM} \to \text{NUM}\}$, and it is axiomatized as follows[1]: $\{\mathsf{s}(x) \simeq \mathsf{s}(y) \to x \simeq y\} \cup \{x \not\simeq \mathsf{s}^n(x) \mid n \in \mathbb{N}, n > 0\}$. T_I is the theory of Integer Offsets defined as $T_S \cup \{\mathsf{s}(x) \not\simeq 0\}$. In the following we will generically denote as T_C any theory in the set $\{T_S, T_I\}$.

In order to deal with theories that are extensions of a theory of Counter Arithmetic we consider the superposition calculus of [18], presented in Figure 1, which extends the standard superposition calculus of [3] to take into account the axioms of the theories T_S or T_I. The difference between the classical calculus and the one we consider in this paper is twofold: (1) this calculus is specialized for reasoning over sets of *literals*, and (2) this calculus is augmented with four rules over ground terms, called Ground Reduction Rules, that encode directly into the calculus the axioms of the theory of Counter Arithmetic.

We will use this calculus to check the satisfiability of any set of ground 'flat' literals modulo a set of axioms. In the context of this paper, a literal is said to be *flat* if it is a Σ_S-literal or a positive literal of depth 1 which is not a Σ_S-literal.

Definition 1. *Let $\mathcal{SP}^{\succ}_I$ be the calculus presented in Figure 1. Let $\mathcal{SP}^{\succ}_S$ be the calculus obtained from $\mathcal{SP}^{\succ}_I$ by removing the rule C1. Let T_C be the generic name for a theory chosen between T_I and T_S and, analogously, let $\mathcal{SP}^{\succ}_C$ be the generic name for a calculus chosen between $\mathcal{SP}^{\succ}_I$ and $\mathcal{SP}^{\succ}_S$.*

The simplification ordering $\succ$ used in the conditions of the rules is total on ground terms.

We now introduce two crucial notions: goodness and safety. The first restricts the choice of a reduction ordering $\succ$ when using the $\mathcal{SP}^{\succ}_C$ calculus. The second is a property of the saturations obtained using $\mathcal{SP}^{\succ}_C$. In [18] it is shown that goodness and safety are sufficient to guarantee that $\mathcal{SP}^{\succ}_C$ is a decision procedure for the satisfiability problem of equational theories extending Counter Arithmetic. In this paper we show that these properties are also sufficient to ensure the modular termination of $\mathcal{SP}^{\succ}_C$ and combinability, when applied to unions of theories.

A simplification ordering $\succ$ which is total on ground terms on a signature containing Σ_S is s-*good* if (1) $t \succ c$ for every ground compound term t which is not s-rooted and every constant c, (2) 0 is minimal, and (3) whenever two terms t_1 and t_2 are not s-rooted we have $\mathsf{s}^m(t_1) \succ \mathsf{s}^n(t_2)$ iff either $t_1 \succ t_2$ or $(t_1 = t_2$ and $m > n)$.

A *derivation* is a sequence $S_0, \ldots, S_i, \ldots$ such that each S_i is a set of literals obtained from S_{i-1} by applying an inference rule to literals in S_{i-1}. We denote S_ω the set of *persistent* literals $\bigcup_i \bigcap_{j>i} S_j$. When the derivation is finite, the set of persistent literals coincides with the last set of the derivation. A derivation is

[1] All the axioms are (implicitly) closed under universal quantification.

fair if whenever an inference is applicable it will be eventually applied unless one of the literals that would be involved in this inference is simplified, subsumed, or deleted (see, e.g. [18] for a formal definition). The set of persistent literals S_ω obtained by a fair derivation is called the *saturation* of the derivation. In the following, we will only consider fair derivations.

Definition 2. *The saturation S_ω of a fair derivation δ w.r.t. $\mathcal{SP}^{\succ}_C$ is* safe *if, whenever S_ω does not contain $\bot$, we have that, for every literal $L \in S_\omega$ and any maximal term t in L: (1) if L is an equality and t is a variable then t is not of sort* NUM*, and (2) if L is an equality and t is* s*-rooted then L is ground (condition (1) is related to* variable inactivity*, see [1]). S_ω is* proper *if in addition, we have that: (3) if t has an* s*-rooted subterm u then the direct subterm of u is a non-variable.*

Definition 3. *Consider an equational Σ-theory T such that $\Sigma \supseteq \Sigma_S$, and assume $\mathcal{SP}^{\succ}_C$ is used with a* s*-*good *ordering $\succ$. The theory T is* terminating *(resp.* safely terminating*/*properly terminating*) w.r.t. $\mathcal{SP}^{\succ}_C$ if, for any set G of ground flat literals (built out of symbols from Σ and possibly further free constants), we have that:*

1. *there exists a saturation S_ω of $T \cup G$ w.r.t. $\mathcal{SP}^{\succ}_C$ which is finite (resp. finite and safe/finite and proper),*
2. *for any set G' of ground Σ_S-literals (built out of symbols from Σ_S and possibly further free constants) such that $G' \cap G = \emptyset$, there exists a saturation of $S_\omega \cup G'$ w.r.t. $\mathcal{SP}^{\succ}_C$ which is finite (resp. finite and safe/finite and proper).*

In this paper, we consider two different ways to find theories satisfying Definition 3. In Section 6, we introduce theories for which all the ground saturations modulo T are of the expected forms. In Section 4, we consider unions of theories for which the saturations described in Definition 3 have the expected forms.

Theorem 1 ([18]). *if T is safely terminating w.r.t. $\mathcal{SP}^{\succ}_C$, then $\mathcal{SP}^{\succ}_C$ induces a decision procedure for the constraint satisfiability problem w.r.t. $T \cup T_C$.*

Note that Definition 3 is slightly stronger than the one of [18]. This is motivated by the assumptions we need to prove modular termination in Section 4.

3.2 Background on Non-Disjoint Combination

Combination techniques are widely studied to build decision procedures for complex theories by using decision procedures for simpler component theories. The Nelson-Oppen method [15] applies to disjoint unions of theories that satisfy stably infiniteness. The combination framework we consider here is an extension of Nelson-Oppen to the non-disjoint case and combines satisfiability procedures having the capability of deducing logical consequences over the shared signature Σ_0. In this framework, the notion of stably infiniteness is generalized by introducing the notions of Noetherianity and T_0-compatibility, where T_0 is the shared theory. In this section we briefly recall these notions.

Expansion Rules:

Superposition	$\dfrac{l[u'] \simeq r \quad u \simeq v}{(l[v] \simeq r)\sigma}$	if (i) and (ii)
Paramodulation	$\dfrac{l[u'] \not\simeq r \quad u \simeq v}{(l[v] \not\simeq r)\sigma}$	if (i) and (ii)
Reflection	$\dfrac{u \not\simeq u'}{\bot}$	

Where the substitution σ is the most general unifier of u and u', and u' is *not* a variable in *Superposition* and *Paramodulation*. Moreover, we have the following conditions: (i) $u\sigma \not\preceq v\sigma$, and (ii) $l[u']\sigma \not\preceq r\sigma$.

Contraction Rules:

Subsumption	$\dfrac{S \cup \{L, L'\}}{S \cup \{L\}}$	if $L\vartheta = L'$ for some substitution ϑ
Simplification	$\dfrac{S \cup \{L[l'], l \simeq r\}}{S \cup \{L[\vartheta(r)], l \simeq r\}}$	if $l' = l\vartheta$, $l\vartheta \succ r\vartheta$, and $L[l\vartheta] \succ (l\vartheta \simeq r\vartheta)$
Deletion	$\dfrac{S \cup \{t \simeq t\}}{S}$	

Ground Reduction Rules:

R1	$\dfrac{S \cup \{\mathsf{s}(u) \simeq \mathsf{s}(v)\}}{S \cup \{u \simeq v\}}$	if u and v are ground
R2	$\dfrac{S \cup \{\mathsf{s}(u) \simeq t, \mathsf{s}(v) \simeq t\}}{S \cup \{u \simeq v, \mathsf{s}(v) \simeq t\}}$	if u, v, and t are ground, and $\mathsf{s}(u) \succ t$, $\mathsf{s}(v) \succ t$, and $u \succ v$
C1	$\dfrac{S \cup \{\mathsf{s}(t) \simeq 0\}}{S \cup \{\mathsf{s}(t) \simeq 0\} \cup \{\bot\}}$	if t is ground
C2	$\dfrac{S \cup \{\mathsf{s}^n(t) \simeq t\}}{S \cup \{\mathsf{s}^n(t) \simeq t\} \cup \{\bot\}}$	if t is ground and $n > 0$

Fig. 1. The superposition calculus $\mathcal{SP}^{\succ}_{C}$ with built-in counter arithmetic.

In order to ensure the termination when deducing logical consequences over the shared signature, we assume that the shared Σ_0-theory is Noetherian. Intuitively, a theory is Noetherian if there exists only a finite number of atoms that are not redundant when reasoning modulo T_0.

Proposition 1 ([18]). *T_C is Noetherian.*

Consider a theory $T \supseteq T_0$ with signatures $\Sigma \supseteq \Sigma_0$. Given an arbitrary set of ground clauses Θ over Σ, the notion of T_0-*basis* allows us to define a 'complete set' of positive logical consequences of Θ over Σ_0.

Definition 4 (T_0-basis). *Given a finite set Θ of ground clauses (built out of symbols from Σ and possibly further free constants) and a finite set of free constants $\underline{a}$, a* T_0-basis *modulo T for Θ w.r.t. $\underline{a}$ is a set Δ of* positive *ground $\Sigma_0^{\underline{a}}$-clauses, denoted by T_0-$\mathsf{basis}_T(\Theta)$, such that*

(i) $T \cup \Theta \models C$, for all $C \in \Delta$ and
(ii) if $T \cup \Theta \models C$ then $T_0 \cup \Delta \models C$, for every positive ground $\Sigma_0^{\underline{a}}$-clause C.

Note that in the definition of a basis we are interested only in positive ground clauses: the exchange of positive information is sufficient to ensure the completeness of the resulting procedure. The interest in Noetherian theories lies in the fact that, for every set of Σ-clauses Θ and for every set $\underline{a}$ of constants, a finite T_0-basis for Θ w.r.t. $\underline{a}$ always exists [10]. Note that if Θ is T-unsatisfiable then w.l.o.g. $\Delta = \{\bot\}$. Unfortunately, a basis for a Noetherian theory does not need to be computable; this motivates the following definition.

Definition 5. *A theory T is an* effectively Noetherian extension of T_0 *if and only if T_0 is Noetherian and a T_0-basis modulo T w.r.t. $\underline{a}$ is computable for every set of literals and every finite set $\underline{a}$ of free constants.*

Theorem 2 ([18]). *Let $\underline{a}$ be a finite set of free constants. Assume $\mathcal{SP}_C^{\succ}$ is used with a* s-good *ordering $\succ$ such that any $\Sigma_S^{\underline{a}}$-term is smaller than any term containing a function symbol not in $\Sigma_S^{\underline{a}}$. If T is safely terminating w.r.t. $\mathcal{SP}_C^{\succ}$, then $\mathcal{SP}_C^{\succ}$ is able to compute a T_C-basis modulo $T \cup T_C$ w.r.t. $\underline{a}$.*

The combination method works by exchanging the shared clauses obtained from procedures computing T_0-bases. To ensure completeness, we rely on the notion of T_0-compatibility. We do not give here a general definition of this notion, but we recall how compatibility instantiates in the particular cases of T_I and T_S.

Proposition 2 ([18]). *A theory T such that $T \supseteq T_I$ is T_I-compatible iff every T-satisfiable constraint is satisfiable in a model of T in which $\forall x(x \not\simeq 0 \Rightarrow \exists y\, x \simeq \mathsf{s}(y))$ holds. A theory T such that $T \supseteq T_S$ is T_S-compatible iff every T-satisfiable constraint is satisfiable in a model of T in which $\forall x\ \exists y\ x \simeq \mathsf{s}(y)$ holds.*

The union of a Σ_1-theory T_1 and a Σ_2-theory T_2 *shares* the Σ_0-theory T_0 if $T_0 \subseteq T_1$, $T_0 \subseteq T_2$, and $\Sigma_1 \cap \Sigma_2 = \Sigma_0$. The following theorem states the modularity result we obtain by applying the Nelson-Oppen combination method extended to unions of theories sharing T_0.

Theorem 3 (Non-disjoint Nelson-Oppen [10]). *Let T_0 be a Noetherian Σ_0-theory. The class of theories which are T_0-compatible and effective Noetherian extensions of T_0 is closed under union sharing T_0.*

In [18], we studied how to apply Theorem 3 when the shared theory is T_C, by considering effectively Noetherian extensions for which $\mathcal{SP}_C^{\succ}$ terminates and is able to compute T_C-bases. This paper is the continuation of [18] with two new contributions. First, we show a modularity result for termination w.r.t. $\mathcal{SP}_C^{\succ}$. Second, we show a general T_C-compatibility result. For these results, we rely on the notions of safe and proper termination w.r.t. $\mathcal{SP}_C^{\succ}$ (see Definition 3).

4 Modular Termination

We are interested in designing a satisfiability procedure for the union of safely terminating theories sharing T_C. Since T_C is Noetherian, an obvious solution could be to use the combination method provided by Theorem 3. In general, this is not an easy task since it requires to prove T_C-compatibility and effective Noetherianity of the component theories. By Theorem 2, we know that whenever the saturation is safe we can infer effective Noetherianity.

As an alternative, we propose a modular termination result that applies to the union of the considered theories and is based on the analysis of the saturations (similarly to [1,4]). This result is interesting in that it does not require us to prove the more complex property of T_C-compatibility for the component theories.

Besides, the application of the combination framework is very useful whenever we consider theories that cannot be easily handled through superposition, such as the theories of arithmetic. By Theorem 2, we know that modular termination entails effective Noetherianity for the union of (non-disjoint) theories. This result satisfies the first requirement of the combination framework. To satisfy the second, we show in Section 5 that the strengthening of safe termination into proper termination is enough to prove compatibility.

In the following, an s-*equality* is a ground equality of the form $a \simeq \mathsf{s}^m(b)$, for some constants a and b of sort NUM.

Lemma 1. *For any theory $T \supseteq T_C$ and any finite set of ground flat literals G, if* (1) *a saturation S_ω of $T \cup G$ by $\mathcal{SP}_C^\succ$ using a s-good ordering does not contain $\bot$ and* (2) *every s-equality in S_ω is either of the form* (i) *$a \simeq \mathsf{s}^m(b)$, for $m \geq 0$ and $a \succ b$, or of the form* (ii) *$\mathsf{s}^m(a) \simeq b$, for $m \geq 1$ and $a \succ b$, then S_ω contains at most one such equality for each pair of distinct constants a, b.*

Proof: By contradiction, assume there are two s-equalities $a \simeq \mathsf{s}^{m_1}(b)$ and $a \simeq \mathsf{s}^{m_2}(b)$ in S_ω with $m_1 \neq m_2$ of the form (i). By superposition in S_ω there is also the literal $\mathsf{s}^{m_1}(b) \simeq \mathsf{s}^{m_2}(b)$, where $m_1 + m_2 > 0$, and $\bot$, generated from that literal by a finite number of applications of rule R1 and an application of rule C2. Let us now assume there are two s-equalities $a \simeq \mathsf{s}^{m_1}(b)$ and $\mathsf{s}^{m_2}(a) \simeq b$ in S_ω of the form (i) and (ii), respectively. Again, by superposition, we would have also $\mathsf{s}^{m_1+m_2}(b) \simeq b$, where $m_1 + m_2 > 0$, and $\bot$, generated from such literal by an application of rule C2. Finally, let us assume there are two s-equalities $\mathsf{s}^{m_1}(a) \simeq b$ and $\mathsf{s}^{m_2}(a) \simeq b$ in S_ω with $m_1 \neq m_2$ and $m_1, m_2 \geq 1$ of the form (ii). This is impossible by fairness, since R2 could have been applied and one of the two would have been deleted. Therefore, we have at most one equality either of the form (i) or of the form (ii) for each pair of constants. □

Theorem 4. *Assume $\mathcal{SP}_C^\succ$ is used with a* s-good *ordering $\succ$. The class of theories which are safely terminating (resp. properly terminating) w.r.t. $\mathcal{SP}_C^\succ$ is closed under union sharing T_C.*

Proof: Let T_i be a safely terminating Σ_i-theory, and G_i be a Σ_i-constraint, for $i = 1, 2$, such that $T_1 \cup T_2$ shares T_C. We first consider the preservation of

termination for $T_1 \cup T_2$, and then we consider the preservation of safety (resp. properness) for $T_1 \cup T_2$.

Termination. Let us consider a fair derivation $\delta = S_0, \ldots, S_k, S_{k+1}, \ldots$ starting from $S_0 = T_1 \cup T_2 \cup G_1 \cup G_2$. We will show (by induction on the length of the derivation) that, for each $S_k \in \delta$: (1) S_k is of the form $S_k^1 \cup S_k^2$ where S_k^i is the set of Σ_i-literals in S_k, for $i = 1, 2$, and (2) any saturation of S_k^i is finite for $i = 1, 2$. By assumption, these two properties hold for $k = 0$. To prove the inductive case, it is sufficient to show that "across-theories" inferences generate only (finitely many) ground shared literals. To do so, we assume that we use a fair strategy that consists in computing the saturations modulo T_1 and T_2 before applying the "across-theories" inferences. Let S_k^1 and S_k^2 be saturated. By assumption, we know that S_k^1 and S_k^2 are safe. Let us analyze superposition and paramodulation inferences (uniformly called paramodulation in the following) from the set S_k^i into the set S_k^j, for $i \neq j$. We have three cases:
(a) Paramodulation from variables. Let $x \simeq t$ be a literal in S_k^i. In order to paramodulate into a literal of S_k^j the variable x must be of sort NUM. By condition (1) of safety, x cannot be maximal in $x \simeq t$ and, therefore, no paramodulation from variables is possible.
(b) Paramodulation from constants. Let $a \simeq t$ be a literal L in S_k^i, where a is a constant. In order to paramodulate from a into a literal in S_k^j, such literal must be of the form $l[a]_p \bowtie r$ and the corresponding m.g.u. is empty. Again, a is of sort NUM. By condition (1) of safety, t can be either a compound term or a constant and, in order to trigger a paramodulation, it must be $a \not\preceq t$. Assume t is compound and of the form $\mathsf{s}^m(u)$ for some $m \geq 0$ and term u which is compound and not s-rooted. Then, by condition (1) of s-goodness, $u \succ a$ and, by the subterm property of $\succ$, $t \succeq u \succ a$, which is a contradiction with the assumption that a is maximal. Thus, we conclude that t is of the form $\mathsf{s}^m(v)$, where v is either a constant or a variable. Now let v be a variable, then also $\mathsf{s}^m(v)$ is maximal and since L is not ground this is not allowed by condition (2) of safety. As a consequence, we have that L is of the form $a \simeq \mathsf{s}^m(b)$, for some constant b, $m \geq 0$, and $a \succ b$.
(c) Paramodulation from compound terms. Let $t \simeq u$ be a literal L in S_k^i and $l \bowtie r$ a literal in S_k^j, where t is a compound term. Since T_1 and T_2 share symbols in a constant expansion of Σ_S, the term t must be s-rooted. Let σ be the m.g.u. of t and $l_{|_p}$, for some position p, then paramodulation requires that $t\sigma \not\preceq u\sigma$. By stability of $\succ$ we have $t \not\preceq u$ as well and t is maximal in $t \simeq u$. By the safety assumption, the literal $t \simeq u$ is ground, σ is the empty substitution, and $t \succ u$. Now, since S_k^i is saturated, u is not s-rooted, otherwise that literal would have been deleted by an application of rule R1. By the safety assumption, L is a ground literal of the form $\mathsf{s}^m(a) \simeq b$, for $m \geq 1$ and $a \succ b$ (the case where $a = b$ is ruled out by the applicability of rule C2).
The literals needed to perform the paramodulation steps of cases (b) and (c) above are s-equalities that satisfy the assumptions of Lemma 1. Hence, by Lemma 1, the set of literals in the saturation of δ that allow a paramodulation in one of the above cases is finite, because it is bounded by n^2, where n is the finite

number of constants occurring in S_0. To formally prove the termination of $\mathcal{SP}^{\succ}_C$, let us consider the following complexity measures for any $S_k \in \delta$: $mse(S_k)$ is n^2 minus the number of s-equalities in S_k, and $mns_i(S_k)$ is the number of remaining steps to reach the saturation of S^i_k, for $i = 1, 2$. We can verify that for any k: (0) $mse(S_k) > mse(S_{k+1})$, or (1) $mse(S_k) \geq mse(S_{k+1})$ and $mns_1(S_k) > mns_1(S_{k+1})$, or (2) $mse(S_k) \geq mse(S_{k+1})$ and $mns_1(S_k) \geq mns_1(S_{k+1})$ and $mns_2(S_k) > mns_2(S_{k+1})$. Therefore, the complexity measure defined as the lexicographic combination of mse, mns_1 and mns_2 is strictly decreased by (each step of) δ, and so δ is finite.

Safe and proper termination. The saturation S_ω of any fair derivation δ is of the form $S^1_\omega \cup S^2_\omega$ such that, for $i = 1, 2$, S^i_ω is a saturation of $T_i \cup G_i \cup E$, where E is the finite set of ground s-equalities generated by δ. By assumption, S^1_ω and S^2_ω are safe (resp. proper), and so $S_\omega = S^1_\omega \cup S^2_\omega$ is safe (resp. proper) too.
The superposition strategy defined for $T_1 \cup T_2$ can be used in an incremental way: given a set G' of new shared literals disjoint from $G_1 \cup G_2$, one can easily check that the saturation of $S_\omega \cup G'$ is still finite since T_1 and T_2 follow Definition 3.□

5 A General Compatibility Result

In this section we show that, by analyzing the saturations, we can infer T_C-compatibility of equational theories extending T_C.

Theorem 5 (T_C-compatibility). *Assume $\mathcal{SP}^{\succ}_C$ is used with a* s-good *ordering $\succ$. If T is properly terminating w.r.t. $\mathcal{SP}^{\succ}_C$, then $T \cup T_C$ is T_C-compatible.*

Proof. By Proposition 2, to show that $T \cup T_C$ is T_C-compatible we have the two cases: (a) for T_I we need to prove that every T-satisfiable constraint S is satisfiable in a model of T where $Pred_I : \forall x(x \not\simeq 0 \Rightarrow \exists y\, x \simeq \mathsf{s}(y))$ holds, and (b) for T_S we need to prove that every T-satisfiable constraint S is satisfiable in a model of T where $Pred_S : \forall x\, \exists y\, x \simeq \mathsf{s}(y)$ holds. We will show that a model $\mathcal{M}$ of $T \cup G$ can be extended to obtain a new model $\mathcal{M}_e$ that satisfies also the above axioms $Pred_I$ and $Pred_S$, respectively (we will refer generically to $Pred_C$).

The construction of the model $\mathcal{M}_e$ from $\mathcal{M}$ is the same for the two theories T_I and T_S, and consists of two steps: (1) we build a sequence $\mathcal{M}_0, \mathcal{M}_1, \ldots$ of models of $T \cup G$, and (2) we define the model $\mathcal{M}_e$ as the *direct limit* [11] of this sequence. Then $\mathcal{M}_e$, by construction, satisfies $T \cup G$ and also the axiom $Pred_C$.

Step 1. We define the sequence $\mathcal{M}_0, \mathcal{M}_1, \ldots$ of models of $T \cup G$ (some of which may be identical) as follows. Each model is constructed starting from a saturated set S^i_ω of unit clauses over a signature Σ_i obtained by extending Σ with a finite set of new constants. Then, we consider the corresponding set $ground(S^i_\omega)$ of all the ground instances of clauses in S^i_ω w.r.t. Σ_i. From $ground(S^i_\omega)$, by using the so-called *model generation* technique [3], we construct a convergent rewriting system $\mathcal{R}_i$ such that, for any pair of ground terms l and r, $S^i_\omega \models l = r$ iff $l{\downarrow_{\mathcal{R}_i}} = r{\downarrow_{\mathcal{R}_i}}$. Finally, the model $\mathcal{M}_i$ is defined as the pair (D_i, F_i), where the domain (or carrier) D_i is the set $T(\Sigma_i)|_{\mathcal{R}_i}$ of $\mathcal{R}_i$-normal forms, and, for every function

symbol $f \in \Sigma_i$, $f_{F_i}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)\downarrow_{\mathcal{R}_i}$ for every $t_1, \ldots, t_n \in D_i$. Obviously $\mathcal{M}_i$ is, by construction, a model of S^i_ω.

We can construct the models $\mathcal{M}_0, \ldots, \mathcal{M}_i, \mathcal{M}_{i+1}, \ldots$ as follows. The set S^0_ω is the set S_ω which is a saturation of $T \cup G$ using $\mathcal{SP}^{\succ}_C$ and Σ_0 is the signature Σ. For $i \geq 0$, if there is a constant c in Σ_i of sort NUM such that $c\downarrow_{\mathcal{R}_i} \neq \mathsf{s}(t)\downarrow_{\mathcal{R}_i}$ for any ground term t and ($T_C = T_S$ or $c\downarrow_{\mathcal{R}_i} \neq 0$), then we define $\Sigma_{i+1} = \Sigma_i \cup \{c'\}$ and $S^{i+1}_\omega = S^i_\omega \cup \{\mathsf{s}(c') = c\}$, where c' is a constant of sort NUM that does not belong to Σ_i, and $c' \succ c$. Otherwise, we define $\Sigma_{i+1} = \Sigma_i$ and $S^{i+1}_\omega = S^i_\omega$. Now, by assumption, there is no occurrence of the term $\mathsf{s}(x)$ in S^i_ω and, since no inference is applicable using the new literal $\mathsf{s}(c') = c$, S^{i+1}_ω is saturated. As a consequence, let φ be a literal in G or a clause in T, for every $i \geq 0$, if $S^i_\omega \models \varphi$ then $S^{i+1}_\omega \models \varphi$. We have also that, for all $0 \leq i < j$, $\mathcal{M}_i \subseteq \mathcal{M}_j$, that is $D_i \subseteq D_j$ and the inclusion map $D_i \to D_j$ is an embedding.

Step 2. We define the limit model $\mathcal{M}_e$ as the pair (D_e, F_e) where the domain is $D_e = \bigcup_{i \geq 0} D_i$ and, for every n-ary function symbol $f \in \Sigma$, and elements $a_1, \ldots, a_n$ of D_e, (by a little abuse of notation) $f_{F_e}(a_1, \ldots, a_n) = f_{F_i}(a_1, \ldots, a_n)$ where $i \geq 0$ is the smallest integer such that $a_1, \ldots, a_n$ are in the domain D_i. By construction, there is an embedding between $\mathcal{M}_i$ and $\mathcal{M}_j$ for $i < j$ (i.e. an injective homomorphism). By (a many-sorted version of) Theorem 2.4.6 in [11] we have that the truth value of a clause φ in $G \cup T$ is *preserved* in $\mathcal{M}_e$, that is, if $\mathcal{M}_0 \models \varphi$ then $\mathcal{M}_e \models \varphi$, and $\mathcal{M}_e$ is a model of $T \cup G$. Furthermore, by construction, there is no constant whose interpretation in $\mathcal{M}_e$ has no predecessor, which entails that also the axiom $Pred_C$ holds.
By these observations we have that the theory T is T_C-compatible. □

The following example shows why we need to strengthen the notion of safe termination into that of proper termination in order to prove T_C-compatibility.

Example 1. Consider the theory $T = \{f(\mathsf{s}(x)) \not\simeq f(c)\}$, which is safely terminating. Let S_ω be a saturation and c be a constant that has no predecessors. If introduce the equality $c \simeq \mathsf{s}(c')$ in S_ω, for some fresh new constant c' not occurring in $T \cup S$. Then, by Paramodulation, we get $f(c) \not\simeq f(c)$ and, thus, the empty clause $\bot$.

6 Applying Modular Termination and Combination

We now consider in more details the function *minmax* introduced in the example of Section 2. We show that the verification task involves theories that are suitable for the application of our modular termination and combination results.

We start by defining the theories of lists T_{LV} and of records T_{RV}. These are enriched with an operator count_D that counts the number of modifications that have been performed on a 'fresh' data structure. Any constant c in the sort of a given data structure D can be declared fresh, i.e. unmodified, by adding the ground literal $\mathsf{count}_D(c) \simeq 0$. Due to the count_D operator, these theories are said to allow 'versioning'. We focus on the constraint satisfiability problem for these theories and their combinations. Through the analysis of their saturations

we show that our modularity results enable us to combine these theories together and with the theory of Linear Rational Arithmetic $T_{\mathbb{Q}}$, so to obtain the satisfiability procedure which is necessary to solve our verification problem.

Lists with versioning. T_{LV} is a theory of lists with extensionality endowed with the counting operator. The many-sorted signature of T_{LV} is given by Σ_S plus the set of function symbols $\{\mathsf{nil} : \textsc{list}, \mathsf{car} : \textsc{list} \to \textsc{elem}, \mathsf{cdr} : \textsc{list} \to \textsc{list}, \mathsf{cons} : \textsc{elem} \times \textsc{list} \to \textsc{list}, \mathsf{count}_L : \textsc{list} \to \textsc{num}\}$ and the predicate symbol $\mathsf{atom} : \textsc{list}$. The axioms of T_{LV} are:

$$\begin{array}{ll} \mathsf{car}(\mathsf{cons}(x,y)) \simeq x & \neg\mathsf{atom}(x) \to \mathsf{cons}(\mathsf{car}(x),\mathsf{cdr}(x)) \simeq x \\ \mathsf{cdr}(\mathsf{cons}(x,y)) \simeq y & \neg\mathsf{atom}(\mathsf{cons}(x,y)) \\ & \mathsf{atom}(\mathsf{nil}) \\ \mathsf{count}_L(\mathsf{cons}(x,y)) \simeq \mathsf{s}(\mathsf{count}_L(y)) & \mathsf{count}_L(\mathsf{nil}) \simeq 0 \end{array}$$

Proposition 3. *T_{LV} is properly terminating w.r.t. $\mathcal{SP}^{\succ}_{C}$.*

Proof. Termination is proved in [18]. In the case of lists, the axioms for the count_L operator are identical up to a renaming to those of the list length operator. Therefore, we can use the analysis of the saturation done in [18]. Any inference generates only ground literals (which do not influence safety/properness). Thus, we can conclude that any saturation S_ω is proper and T_{LV} is properly terminating. □

Records with versioning. T_{RV} is theory of records with extensionality endowed with the counting operator. The many-sorted signature of T_{RV} is given by Σ_S and the function symbols defined as follows. Let RECORD be the sort of records; for every attribute identifier we have two functions $\mathsf{rselect_i} : \textsc{record} \to \textsc{elem}_i$ and $\mathsf{rstore_i} : \textsc{record} \times \textsc{elem}_i \to \textsc{record}$, where $1 \leq i \leq n$. Moreover, there is the function $\mathsf{count}_R : \textsc{record} \to \textsc{num}$ that counts the number of $\mathsf{rstore_i}$ operations performed on a record. The axioms of T_{RV} are (for every i, j such that $1 \leq i, j \leq n$ and $i \neq j$):

$$\begin{array}{rl} & \mathsf{rselect_i}(\mathsf{rstore_i}(x,y)) \simeq y \\ & \mathsf{rselect_j}(\mathsf{rstore_i}(x,y)) \simeq \mathsf{rselect_j}(x) \\ \wedge_{i=1}^{n}(\mathsf{rselect_i}(x) \simeq \mathsf{rselect_i}(y)) \to x \simeq y & \qquad \text{(extensionality)} \\ & \mathsf{count}_R(\mathsf{rstore_i}(x,y)) \simeq \mathsf{s}(\mathsf{count}_R(x)) \end{array}$$

By using the reduction described in [1], which is valid also with the additional axioms for the function count_R, it is possible to drop the extensionality axiom, so that the theory of records is *equational*. As a consequence, we restrict our attention to (equisatisfiable) sets of literals in which no disequation between records appears and we focus on the saturation of sets of literals of the forms:

i. equational axioms for records:
   a. $\mathsf{rselect_i}(\mathsf{rstore_i}(x,y)) \simeq y$, b. $\mathsf{rselect_j}(\mathsf{rstore_i}(x,y)) \simeq \mathsf{rselect_j}(x)$,
   c. $\mathsf{count}_R(\mathsf{rstore_i}(x,y)) \simeq \mathsf{s}(\mathsf{count}_R(x))$;

ii. ground literals over the sorts RECORD and ELEM_i, for $i \in \{1, \ldots, n\}$:
a. $r \simeq r'$, b. $e \simeq e'$, c. $e \not\simeq e'$, d. $\mathsf{rselect_i}(r) \simeq e$, e. $\mathsf{rstore_i}(r, e) \simeq r'$;
iii. ground literals over the sort NUM:
a. $\mathsf{count}_R(r) \simeq \mathsf{s}^n(k)$, b. $\mathsf{s}^n(k) \simeq k'$, c. $\mathsf{s}^m(k) \not\simeq \mathsf{s}^n(k')$;

where e, e' are constants of sort ELEM_i, r, r' are constants of sort RECORD, and k, k' are constants of sort NUM. Note that 0 is one of the constants of sort NUM and, thus, in case (iii.a) is included also the literal $\mathsf{count}_R(r) \simeq 0$.

We consider an LPO ordering $\succ$ over terms such that the underlying precedence over the symbols in the signature satisfies the following requirements: for all i, j in $\{1, \ldots, n\}$, $\mathsf{rstore_i} > \mathsf{rselect_j}$, $\mathsf{rstore_i} > \mathsf{count}_R$, $\mathsf{rselect_i} > c$, and $\mathsf{count}_R > c > 0 > \mathsf{s}$, for every constant c.

Proposition 4. *T_{RV} is properly terminating w.r.t. $\mathcal{SP}^{\succ}_{C}$.*

Proof. Let us consider a set S_0 of literals of the form (i)–(iii). We prove that any saturation S_ω of S_0 constructed using $\mathcal{SP}^{\succ}_{C}$ is finite and proper. Any literal introduced by an inference rule is ground and smaller that the biggest literal in the input set. By well-foundedness of the multiset extension of the (well-founded) ordering $\succ$ we get termination. Since the saturation generates only ground literals (which do not affect safety/properness), the analysis of (i)–(iii) is sufficient to conclude that S_ω is proper and T_{RV} is properly terminating. □

Corollary 1. *$T_{LV} \cup T_{RV}$ is properly terminating w.r.t. $\mathcal{SP}^{\succ}_{C}$, $\mathcal{SP}^{\succ}_{C}$ computes a T_C-basis modulo $T_{LV} \cup T_{RV} \cup T_C$, and $T_{LV} \cup T_{RV} \cup T_C$ is T_C-compatible.*

Proof. By Propositions 3 and 4, Theorem 4, and Theorem 5. □

Theory of Linear Rational Arithmetic. $T_{\mathbb{Q}}$ is the theory of Linear Rational Arithmetic discussed in [18], whose signature over the sort NUM is $\Sigma_{\mathbb{Q}} := \{0, 1, +, -, \mathsf{s}, <\}$, where $0, 1$ are constants, $-$ and s are unary function symbols, $+$ is a binary function symbol and $<$ is a binary predicate symbol. The symbols $0, 1, +, -, \mathsf{s}, <$ are interpreted in their intended meaning. In particular, s is the function that associates to each rational q the rational $q + 1$. Clearly, $T_S \subseteq T_{\mathbb{Q}}$. In [18], it is shown that all the assumptions of the Combination Theorem (Theorem 3) are fulfilled for $T_{\mathbb{Q}}$ when $T_0 = T_S$.

As a consequence of our modular termination result we have that $\mathcal{SP}^{\succ}_{S}$ is a satisfiability procedure for $T_{LV} \cup T_{RV} \cup T_S$ and it can be used to compute a T_S-basis modulo $T_{LV} \cup T_{RV} \cup T_S$. Since $T_{LV} \cup T_{RV} \cup T_S$ is also T_S-compatible, all the assumptions of the Combination Theorem (Theorem 3) are satisfied. Hence, we can construct a combined satisfiability procedure for $(T_{LV} \cup T_{RV} \cup T_S) \cup T_{\mathbb{Q}}$ by combining the $\mathcal{SP}^{\succ}_{S}$ calculus, used as a satisfiability procedure for $(T_{LV} \cup T_{RV} \cup T_S)$, and a satisfiability procedure for $T_{\mathbb{Q}}$.

Example 2. Let us consider the function *minmax* given in Section 1. Assume we want to prove that the record r is not modified more than 'size of l' times. This amounts to prove that the formula $\psi : \ \forall l, r\, (r = minmax(l) \Rightarrow \mathsf{count}_R(r) \leq \mathsf{count}_L(l))$ holds. In order to prove ψ we need to prove, for the loop invariant, the

formula β: $\forall(\gamma \Rightarrow (\mathsf{count}_R(r) \leq n - \mathsf{count}_L(l) \Rightarrow \mathsf{count}_R(r'') \leq n - \mathsf{count}_L(l')))$ where γ is the conjunction of:

$\mathsf{rselect}_1(r) \leq \mathsf{rselect}_2(r)$ $\quad$ $\mathsf{rselect}_2(r') < e \Rightarrow r'' \simeq \mathsf{rstore}_2(r', e)$
$e < \mathsf{rselect}_1(r) \Rightarrow r' \simeq \mathsf{rstore}_1(r, e)$ $\quad$ $e \leq \mathsf{rselect}_2(r') \Rightarrow r'' \simeq r'$
$\mathsf{rselect}_1(r) \leq e \Rightarrow r' \simeq r$ $\quad$ $l' \simeq \mathsf{cdr}(l)$

Note that, without constraints on the initial values of r the record can be updated more than 'size of l' times. This motivates the assumption $\mathsf{rselect}_1(r) \leq \mathsf{rselect}_2(r)$. The formula β is over the signature $\Sigma_{LV} \cup \Sigma_{RV} \cup \Sigma_{\mathbb{Q}}$ and its validity can be proved by using a SMT solver modulo $T_{LV} \cup T_{RV} \cup T_S \cup T_{\mathbb{Q}}$. Applying our results about the superposition calculus $\mathcal{SP}^{\succ}_S$ together with a combination procedure for the shared theory T_S, we can obtain the necessary satisfiability procedure for $T_{LV} \cup T_{RV} \cup T_S \cup T_{\mathbb{Q}}$.

7 Conclusions

In this paper, we have identified the key notion of safe termination that allows us to prove modular termination and completeness (modulo T_C) of the superposition calculus and the combination framework, respectively. In particular, we have shown that safe termination implies modular termination and proper termination implies compatibility. In the signature-disjoint case, variable-inactivity has been initially introduced to obtain modular termination [1] but it is useful for combinability as well [12] to ensure: (1) that the bases are computable (deduction completeness) and (2) stably infiniteness. In the non-disjoint case, compatibility replaces stably infiniteness. Through these results we show an analogy between variable inactivity and safety. Roughly speaking, safety replaces variable inactivity when considering (unions of) theories sharing T_C.

The property of safe termination of saturation has to be verified for any given set of ground flat literals. Meta-superposition [12,13] has proved to be useful, in the disjoint case, for checking termination and variable inactivity on a single schematic form of saturation. We plan to develop a meta-superposition calculus modulo counter arithmetics to perform an automatic check of termination and safety.

A further research direction is the extension of our superposition calculus to non-convex theories (that is, non-Horn theories). In that case, the calculus would require significant changes in order to obtain completeness and an effective way to compute the bases, containing entailed *disjunctions* of s-equalities.

References

1. A. Armando, M. P. Bonacina, S. Ranise, and S. Schulz. New results on rewrite-based satisfiability procedures. *ACM Trans. Comput. Log.*, 10(1), 2009.
2. A. Armando, S. Ranise, and M. Rusinowitch. A rewriting approach to satisfiability procedures. *Inf. Comput.*, 183(2):140–164, 2003.

3. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *J. Log. Comput.*, 4(3):217–247, 1994.
4. M. P. Bonacina and M. Echenim. On Variable-inactivity and Polynomial T-Satisfiability Procedures. *J. Log. Comput.*, 18(1):77–96, 2008.
5. M. P. Bonacina and M. Echenim. Theory decision by decomposition. *J. Symb. Comput.*, 45(2):229–260, 2010.
6. M. P. Bonacina, C. Lynch, and L. M. de Moura. On Deciding Satisfiability by $DPLL(\Gamma + \mathcal{T})$ and Unsound Theorem Proving. In Schmidt [19], pages 35–50.
7. N. Dershowitz and D. Plaisted. Rewriting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 9, pages 535–610. Elsevier Science, 2001.
8. H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York-London, 1972.
9. S. Ghilardi. Model-theoretic methods in combined constraint satisfiability. *J. Autom. Reasoning*, 33(3-4):221–249, 2004.
10. S. Ghilardi, E. Nicolini, and D. Zucchelli. A comprehensive combination framework. *ACM Trans. Comput. Log.*, 9(2), 2008.
11. W. Hodges. *Model Theory*. Number 42 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993.
12. H. Kirchner, S. Ranise, C. Ringeissen, and D.-K. Tran. Automatic combinability of rewriting-based satisfiability procedures. In *LPAR 2006, Proceedings*, volume 4246 of *LNCS*, pages 542–556. Springer, 2006.
13. C. Lynch and D.-K. Tran. Automatic decidability and combinability revisited. In *Automated Deduction - CADE-21, Proceedings*, volume 4603 of *LNCS*, pages 328–344. Springer, 2007.
14. Z. Manna, H. B. Sipma, and T. Zhang. Verifying balanced trees. In *LFCS 2007, Proceedings*, volume 4514 of *LNCS*, pages 363–378. Springer, 2007.
15. G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, 1979.
16. E. Nicolini, C. Ringeissen, and M. Rusinowitch. Combinable extensions of abelian groups. In Schmidt [19], pages 51–66.
17. E. Nicolini, C. Ringeissen, and M. Rusinowitch. Data structures with arithmetic constraints: A non-disjoint combination. In *FroCoS 2009, Proceedings*, volume 5749 of *LNCS*, pages 319–334. Springer, 2009.
18. E. Nicolini, C. Ringeissen, and M. Rusinowitch. Combining satisfiability procedures for unions of theories with a shared counting operator. *Fundam. Inform.*, 105(1-2):163–187, 2010.
19. R. A. Schmidt, editor. *Automated Deduction - CADE-22, Proceedings*, volume 5663 of *LNCS*. Springer, 2009.
20. V. Sofronie-Stokkermans. Locality results for certain extensions of theories with bridging functions. In Schmidt [19], pages 67–83.
21. P. Suter, M. Dotta, and V. Kuncak. Decision procedures for algebraic data types with abstractions. In *Proceedings of the 37th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2010*, pages 199–210. ACM, 2010.