UNIVERSITÀ DEGLI STUDI DI MILANO

Dipartimento di Scienze dell'Informazione

RAPPORTO INTERNO N° 308-06

Decidability and Undecidability Results for Nelson-Oppen and Rewrite-based Decision Procedures

Maria Paola Bonacina, Silvio Ghilardi, Enrica Nicolini,
Silvio Ranise, Daniele Zucchelli

Decidability and Undecidability Results for Nelson-Oppen and Rewrite-based Decision Procedures

Maria Paola Bonacina[1], Silvio Ghilardi[2], Enrica Nicolini[3], Silvio Ranise[2,4], and Daniele Zucchelli[2,4]

[1]Dipartimento di Informatica - Università degli Studi di Verona (Italia)
[2]Dipartimento di Informatica - Università degli Studi di Milano (Italia)
[3]Dipartimento di Matematica - Università degli Studi di Milano (Italia)
[4]LORIA& INRIA-Lorraine, Nancy (France)

March 6, 2006

Abstract

In the context of combinations of theories with disjoint signatures, we classify the component theories according to the decidability of constraint satisfiability problems in finite and infinite models, respectively. We exhibit a theory T_1 such that satisfiability is decidable, but satisfiability in infinite models is undecidable. It follows that satisfiability in $T_1 \cup T_2$ is undecidable, whenever T_2 has only infinite models, even if signatures are disjoint and satisfiability in T_2 is decidable.

In the second part of the paper we strengthen the Nelson-Oppen decidability transfer result, by showing that it applies to theories over disjoint signatures, whose satisfiability problem, in either finite or infinite models, is decidable. We show that this result covers decision procedures based on rewriting, generalizing recent work on combination of theories in the rewrite-based approach to satisfiability.

1 Introduction

We investigate the requirement of being stably-infinite for a (decidable) theory to be combined with others, by using the well-known Nelson-Oppen combination schema. Recently, relaxing this requirement has received a lot of attention in order to design combination schemas handling theories that are not stably-infinite. For instance,[5] Tinelli and Zarba [26] have shown how to combine an arbitrary theory with one satisfying requirements which are stronger than stable-infiniteness. Thus, contrary to the combination schema by Nelson-Oppen [16], such a schema is asymmetric in the sense that the requirements on the component theories are not the same.

[5]For lack of space, we only discuss results which are closely related to ours (see, e.g., [23] for an overview on combination of decision procedures and references).

In this paper, we consider combinations of theories whose signatures are disjoint and classify the component theories according to the decidability of their satisfiability problems in finite and infinite models, respectively (notice that such problems coincide for stably-infinite theories). Assume that the satisfiability problem in a theory T_1 is decidable in arbitrary models but not in infinite models. Then, any combination of such a T_1 with a theory T_2 that does not have finite models yields an undecidable satisfiability problem. This holds even if T_1 and T_2 have disjoint signatures and even if satisfiability in T_2 is decidable in arbitrary models. As a consequence of this observation, we obtain the first (undecidability) result of the paper, by exhibiting a theory such that the satisfiability problem is decidable, whereas the satisfiability problem in infinite models is undecidable.

The second result of the paper is related to decision procedures based on rewriting. Armando et al [1] recently showed how to use a rewrite-based inference system to obtain decision procedures for (disjoint) unions of *variable-inactive* theories, when there exist rewrite-based decision procedures for the component theories. Here, we explain the relationship between variable-inactivity and stable-infiniteness. We show that if a theory is not stably infinite, then the inference system is guaranteed to generate clauses that constrain the cardinality of its models, so that the theory is not variable-inactive. This result has two applications: first, it generalizes the combination schema of [1] for (disjoint) unions of theories that have a rewrite-based satisfiability procedures. Second, it suggests a simple way to combine the rewrite-based approach with constraint-solving techniques that check satisfiability in finite models.

2 Preliminaries

A *signature* Σ is an (at most countable) set of functions and predicate symbols, each of them endowed with the corresponding arity. We assume the binary equality predicate symbol '=' to be always present in any signature Σ. The signature obtained from Σ by the addition of a set of new constants (that is, 0-ary function symbols) $\mathcal{K}$ is denoted by $\Sigma \cup \mathcal{K}$ or by $\Sigma^{\mathcal{K}}$; when the set of constants is finite, we use letters $\underline{a}, \underline{b}, \underline{c}$, etc. in place of $\mathcal{K}$. We have the usual notions of Σ-*term*, (full first order) -*formula*, -*atom*, -*literal*, -*clause*, -*positive clause*, etc.: e.g., an atom is an atomic formula, a literal is an atom or the negation of an atom, a clause is a multiset of literals, a positive clause is a multiset of atoms, etc. Abusing notation, we write a clause C either as the disjunction of its literals or as a sequent $\Delta_1 \Rightarrow \Delta_2$, meaning that Δ_1 (resp. Δ_2) contains the negative (resp. positive) literals of C. Terms, literals, clauses and formulæ are called *ground* whenever variables do not appear. Formulæ without free variables are called *sentences*. The universal (resp. existential) closure of a formula ϕ is the sentence obtained from ϕ by adding a prefix of universal (resp. existential) quantifiers binding all variables occurring free in ϕ. A Σ-*theory* T is a set of sentences (called the axioms of T) in the signature Σ. If T is finite, the theory is said to be finitely axiomatized. A *universal* theory is a theory whose axioms are universal closures of quantifier-free sentences.

From the semantic side, we have the standard notion of a Σ-*structure* $\mathcal{A}$: this is a support set endowed with an arity-matching interpretation of the function and predicate symbols from Σ. We use $f^{\mathcal{A}}$ (resp. $P^{\mathcal{A}}$) to denote the interpretation of the function symbol f (resp. predicate symbol

P) in the structure $\mathcal{A}$. The support set of a structure $\mathcal{A}$ is indicated by the notation $|\mathcal{A}|$. We say that $\mathcal{A}$ is *finite* when there exists an integer $N > 0$ such that the cardinality of $|\mathcal{A}|$ is less than N; if such an integer does not exist, we say that $\mathcal{A}$ is *infinite*. The *truth* of a Σ-formula in $\mathcal{A}$ is defined in the standard way (so that truth of a formula is equivalent to truth of its *universal* closure). A formula ϕ is *satisfiable* in $\mathcal{A}$ iff its *existential* closure is true in $\mathcal{A}$.

A Σ-structure $\mathcal{A}$ is a *model* of a Σ-theory T (in symbols $\mathcal{A} \models T$) iff all axioms of T are true in $\mathcal{A}$. For models of a Σ-theory T we shall use the letters $\mathcal{M}, \mathcal{N}, \ldots$ to distinguish them from arbitrary Σ-structures. If ϕ is a formula, $T \models \phi$ ('ϕ *is a logical consequence of* T') means that ϕ is true in any model of T. A Σ-theory T is *complete* iff for every Σ-sentence ϕ, either ϕ or $\neg\phi$ is a logical consequence of T; T is *consistent* iff it has a model.

A Σ-*constraint* in a signature Σ is a finite set of ground $\Sigma^{\underline{a}}$-literals (where $\underline{a}$ is a finite set of new free constants); the *constraint satisfiability problem* for a Σ-theory T is the problem of deciding a Σ-constraint is satisfiable in a model of T: if this problem is decidable, we say that the theory T is $\exists$-*decidable*. Notice that, equivalently, T is $\exists$-decidable iff it is decidable whether a universal Σ-formula is entailed by the axioms of T.

3 Satisfiability in Finite and Infinite Models

Let T_1 and T_2 be theories such that the signature Σ_1 of T_1 is disjoint from the signature Σ_2 of T_2, i.e., $\Sigma_1 \cap \Sigma_2$ contains only the equality symbol. We consider the decidability of the constraint satisfiability problem of the theory $T_1 \cup T_2$. We are especially interested in establishing the relationships between the decidability of the constraint satisfiability problems in the component theories T_1 and T_2, and the decidability of the constraint satisfiability problem in $T_1 \cup T_2$.

3.1 Undecidability Result

Let us recall two simple facts. First, combined word problems are decidable whenever the word problems for the component theories are decidable [21]. Second, it is commonly believed that combining word problems is more difficult than combining constraint satisfiability problems - the reason is that the algorithms to be combined are less powerful, as they can handle only constraints formed by a single negative literal. From these two observations, one may conjecture that the decidability of the constraint satisfiability problem in $T_1 \cup T_2$ always follows from the decidability of the constraint satisfiability problem in T_1 and T_2. Contrary to expectation, all known combination results for the decidability of the constraint satisfiability problems in unions of theories (such as [16, 26]) assume that the component theories satisfy certain requirements. The key observation is that such requirements are related to the satisfiability of constraints in infinite models of a component theory. For example, the Nelson-Oppen combination schema [16] requires the component theories to be stably-infinite. A Σ-theory T is *stably infinite* iff every Σ-constraint satisfiable in a model of T is satisfiable in an infinite model of T. Motivated by this observation, we introduce the following definition.

Definition 3.1. *Let T be a Σ-theory.*

– T *is* $\exists$-decidable *iff it is decidable whether any* Σ*-constraint* Γ *is satisfiable in an* arbitrary *model of* T*;*

– T *is* $\exists_\infty$-decidable *iff it is* $\exists$*-decidable and moreover it is decidable whether any* Σ*-constraint* Γ *is satisfiable in an* infinite *model of* T*.*

Notice that for stably infinite theories $\exists$-decidability is equivalent to $\exists_\infty$-decidability. To illustrate the interest of studying the decidability of satisfiability in the infinite models of a theory, we state the following

Theorem 3.2. *Let* T_i *be a* Σ_i*-theory (for* $i = 1, 2$*) and let the signatures* Σ_1, Σ_2 *be disjoint. If* T_1 *is* $\exists$*-decidable but it is not* $\exists_\infty$*-decidable and if* T_2 *is consistent,* $\exists$*-decidable but does not admit finite models, then the constraint satisfiability for* $T_1 \cup T_2$ *is undecidable.*

Proof. We simply show that a Σ_1-constraint Γ is $T_1 \cup T_2$-satisfiable iff it is satisfiable in an infinite model of T_1. One side is obvious; for the other side, pick infinite models $\mathcal{M}_1$ of $T_1 \cup \Gamma$ and $\mathcal{M}_2$ of T_2 (the latter exists by consistency of T_2). By Löwhenheim-Skolem theorem, we can assume that both models are countable, i.e. that they have the same support (up to isomorphism). But then, we can simply put together the interpretations of functions and predicate symbols and get a model of $T_1 \cup T_2 \cup \Gamma$. □

We notice that there are many theories which are $\exists$-decidable and have only infinite models. One such theory is Presburger Arithmetic [22], another one is the theory of acyclic lists [20]. More interestingly, one could ask the following

QUESTION 1: Are there $\exists$*-decidable that are not* $\exists_\infty$*-decidable?*

If the answer is positive, then Theorem 3.2 implies that *there exist theories which are* $\exists$*-decidable and whose union is not* $\exists$*-decidable.* In Section 4, we exhibit some theories that are $\exists$-decidable but not $\exists_\infty$-decidable, thereby answering *QUESTION 1* positively.

3.2 Decidability Result

Notwithstanding the negative result implied by Theorem 3.2, we observe that when both T_1 and T_2 are $\exists_\infty$-decidable, we are close to get the decidability of constraint satisfiability in $T_1 \cup T_2$. To understand why, recall the following well-known fact.

Lemma 3.3. *Let* Λ *be a set of first-order sentences. If* Λ *does not admit infinite models, then there must exist an integer* $N > 0$ *such that, for each model* $\mathcal{M}$ *of* Λ*, the cardinality of the support set of* $\mathcal{M}$ *is bounded by* N*.*

For a proof, the interested reader is referred to any introductory textbook about model theory (see, e.g., [27]). The key idea is to apply compactness to infinitely many 'at-least-n-elements' constraints (these are the constraints expressed by the formulæ $\exists x_1, \ldots, x_n \bigwedge_{i \neq j} x_i \neq x_j$). It is interesting to notice that the above bound on the cardinality of finite models can be effectively computed for $\exists$-decidable theories.

Lemma 3.4. *Let T be an $\exists$-decidable Σ-theory; whenever it happens*[6] *that a given Σ-constraint Γ is not satisfiable in an infinite model, one can compute a natural number N such that all models of $T \cup \Gamma$ have cardinality at most N.*

Proof. For $h = 2, 3, \ldots$, add the following set $\delta_h := \{c_i \neq c_j \mid 1 \leq i < j \leq h\}$ of literals to $T \cup \Gamma$, where the constants $c_1, \ldots, c_h$ are fresh.[7] Clearly, if $T \cup \Gamma \cup \delta_h$ is unsatisfiable, then we get a bound for the cardinality of the models of $T \cup \Gamma$. Since, by Lemma 3.3, such a bound exists, the process eventually terminates. □

Definition 3.5. *An $\exists_\infty$-decidable Σ-theory T is said to be* strongly $\exists_\infty$*-decidable iff for any finite Σ-structure $\mathcal{A}$, it is decidable whether $\mathcal{A}$ is a model of T.*

It is not difficult to find strongly $\exists_\infty$-decidable theories. For example, any finitely axiomatizable $\exists_\infty$-decidable Σ-theory with a finite Σ is strongly $\exists_\infty$-decidable, since it is sufficient to check the truth of the axioms for finitely many valuations. Now, we are in the position to state and prove the following modularity property for $\exists_\infty$-decidable theories.

Theorem 3.6. *Let T_i be a strongly $\exists_\infty$-decidable Σ_i-theory (for $i = 1, 2$) such that Σ_1, Σ_2 are finite and disjoint. Then the combined theory $T_1 \cup T_2$ is $\exists$-decidable.*[8]

Proof. Let Γ be a finite set of ground $\Sigma_1 \cup \Sigma_2$-literals containing free constants. By well-known means (see, e.g., [5]), we can obtain an equisatisfiable set $\Gamma_1 \cup \Gamma_2$ such that Γ_i contains only $\Sigma_i^{\underline{a}}$-symbols, for $i = 1, 2$ and for some free constants $\underline{a}$. Let Γ_0 be an *arrangement* of the constants $\underline{a}$, i.e. a finite set of literals such that either $a_i = a_j \in \Gamma_0$ or $a_i \neq a_j \in \Gamma_0$, for $i \neq j$ and $a_i, a_j \in \underline{a}$. Clearly, $\Gamma_1 \cup \Gamma_2$ is satisfiable iff $\Gamma_1 \cup \Gamma_0 \cup \Gamma_2$ is satisfiable for some arrangement Γ_0 of the constants $\underline{a}$. From the fact that theories T_1, T_2 are both $\exists_\infty$-decidable, the following case analysis can be *effectively* performed:

- If $\Gamma_0 \cup \Gamma_i$ is satisfiable in an infinite model of T_i (for both $i = 1, 2$), then $\Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ is satisfiable in an infinite model of $T_1 \cup T_2$ by the standard argument underlying the correctness of the Nelson-Oppen combination schema (see, e.g., [25, 13]).
- If $\Gamma_0 \cup \Gamma_i$ is unsatisfiable in any infinite model of T_i (for either $i = 1$ or $i = 2$), then (by Lemma 3.4) we can effectively compute an integer $N > 0$ such that each model $\mathcal{M}$ of $T \cup \Gamma_i \cup \Gamma_0$ has cardinality less than N. Hence, it is sufficient to exhaustively search through $\Sigma_1 \cup \Sigma_2 \cup \underline{a}$-structures up to cardinality N. The number of these structures is finite because Σ_1 and Σ_2 are finite and, by Definition 3.5, it is possible to effectively check whether each such a structure is a model of T_1 and T_2, and hence also of $T_1 \cup T_2 \cup \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$. If a model is found, the procedure returns 'satisfiable', otherwise another arrangement Γ_0 (if any) is tried.

□

[6] There is a subtle point here: Lemma 3.4 applies to all $\exists$-decidable theories, but it is really useful only for $\exists_\infty$-decidable theories, because only for these theories the hypothesis 'Γ in not satisfiable in an infinite model of T' can be effectively checked.

[7] Notice that the literals in δ_h are simply the Skolemization of the 'at-least-h-elements' constraint.

[8] This result can be easily generalized to the combination of $n > 2$ theories.

Theorem 3.6 raises the following

QUESTION 2: Is there a practical sufficient condition for a theory to be strongly $\exists_\infty$-decidable?

Clearly, stably infinite $\exists$-decidable theories are $\exists_\infty$-decidable. More interesting examples are given in Section 5, where we will show that, whenever a finitely axiomatized theory T admits a rewrite-based decision procedure for its constraint satisfiability problem [2, 1], T is not only $\exists$-decidable but also strongly $\exists_\infty$-decidable.

4 Undecidability

In this section, we give an affirmative answer to *QUESTION 1* by defining some $\exists$-decidable theories that are not $\exists_\infty$-decidable. Let Σ_{TM_∞} be the signature containing (in addition to the equality predicate) the following (infinite) set of propositional letters $\{P_{(e,n)} \mid e, n \in \mathbb{N}\}$. Consider the propositional letter $P_{(e,n)}$: we regard e as the index (i.e. the code) of a Turing Machine and n as the input to the Turing machine identified by e (this coding is possible because of basic results about Turing machines, see, e.g., [19]). We indicate by $k : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ the (non-computable) function associating to each pair (e, n) the number $k(e, n)$ of computation steps of the Turing Machine e on the input n. We write $k(e, n) = \infty$ when the computation does not halt. The axioms of the theory TM_∞ are the universal closures of the following formulæ:

$$P_{(e,n)} \rightarrow \bigvee_{i<j\leq m} x_i = x_j, \qquad \text{if } k(e, n) < m. \tag{1}$$

Two observations are in order. First, the property "being an axiom of TM_∞" is decidable, because the ternary predicate $k(e, n) < m$ is recursive. Indeed, it is sufficient to run the Turing Machine e on input n and wait at most m computation steps to verify whether e halts. Second, the consequent of implication (1) is an *at-most cardinality constraint*, i.e. it is a formula of the form

$$\bigvee_{i\neq j} x_i = x_j \tag{2}$$

where x_i, x_j are (implicitly universally quantified) distinct variables for $i, j = 1, \ldots, n$, which constrain the domain of any model to contain at most n elements. Thus, axioms of the form (1) tells us that if $P_{(e,n)}$ holds and the the Turing Machine e halts in at most m steps, then the cardinality of the domains of a model is bounded by m. These properties allow us to state and prove the following key result:

Proposition 4.1. *The theory TM_∞ is $\exists$-decidable but it is not $\exists_\infty$-decidable.*

Proof. To show that the theory is $\exists$-decidable, consider a constraint Γ over the signature $\Sigma^{\underline{a}}_{TM_\infty}$. First, guess an arrangement Γ_0 for the constants $\underline{a}$ and check the set of equations and inequations from $\Gamma \cup \Gamma_0$ for consistency in the pure theory of equality. Then, if the satisfiability check succeeds, Γ_0 explicitly gives the minimum cardinality m for $\Gamma \cup \Gamma_0$ to be satisfied. Clearly, $\Gamma \cup \Gamma_0$ is unsatisfiable if it contains both $P_{(e,n)}$ and $\neg P_{(e,n)}$. If this is not the case, we still have to consider

the constraints represented by axiom (1), which states that if a literal of the kind $P_{(e,n)}$ is in a Σ_{TM_∞}-constraint, such a constraint can be only satisfied in a model whose cardinality is at most $k(e,n)$. Thus, if $P_{(e,n)} \in \Gamma \cup \Gamma_0$, we only need to check that $m \leq k(e,n)$, which can be effectively done since the ternary predicate $k(e,n) < m$ is recursive.

To see that TM_∞ is not $\exists_\infty$-decidable, notice that the constraint $\{P_{(e,n)}\}$ is TM_∞-satisfiable in an infinite structure iff $k(e,n) = \infty$. In turn, this is equivalent to check whether the computation of the Turing Machine e on the input n does not terminate, which is obviously undecidable, being the complement of the Halting problem. □

The theory TM_∞ is defined on an infinite signature. However, it is possible to introduce two theories TM_ω and $TM_{\forall\omega}$ over finite signatures, with the same characteristics as TM_∞ as far as decidability in finite and infinite models is concerned, and such that $TM_{\forall\omega}$ is also universal. Since the proofs that such theories are $\exists$-decidable but not $\exists_\infty$-decidable are similar to that of Proposition 4.1, modulo some technical details, we report their development in Appendix A. Thus, we are ready to state our first main result:

Theorem 4.2. *There exist two $\exists$-decidable universal theories over finite and disjoint signatures, whose union is not $\exists$-decidable.*

This result follows from Theorem 3.2 and the fact that $TM_{\forall\omega}$ is $\exists$-decidable but not $\exists_\infty$-decidable (cf. Proposition A.2 in Appendix A).

5 Decidability

The answer to *QUESTION 2* rests on showing that (under suitable assumptions) rewrite-based methods give practical sufficient conditions for a theory to be strongly $\exists_\infty$-decidable. First, we need to introduce some technical definitions. In Section 5.1, we recall some basic notions underlying the superposition calculus [18] and we introduce superposition modules as suitable abstractions for the subsequent technical development. Then, in Section 5.2, we introduce the notion of invariant superposition modules and, in Section 5.3, we show that they can generate an "at most" cardinality constraint (cf. (2) in Section 4) whenever a theory does not admit infinite models. Last, in Section 5.4, we describe how to combine rewrite-based procedures [1, 2] with Satisfiability Modulo Theory (SMT) tools, such as [10, 3, 11, 12], in order to obtain automatic methods to solve constraint satisfiability problems involving theories admitting only finite models (e.g., enumerated data-types).

5.1 Superposition Calculi and Superposition Modules

From now on, we consider only universal, finitely axiomatized theories, whose signatures are finite. Without loss of generality, we assume that signatures contain only function symbols.[9] A fundamental assumption of superposition-based inference systems [18] is that the universe of terms

[9] Any atom $P(t_1, \ldots, t_n)$ with predicate symbol P other than equality can be written as an equation $p(t_1, \ldots, t_n) = \mathsf{true}$, where p is a fresh function symbol and true a fresh constant symbol. This transformation preserves satisfiability (see, e.g., [18]).

is ordered by a *reduction ordering*. A reduction ordering on terms can be extended to literals and clauses by using standard techniques. The most commonly used orderings are the *Knuth-Bendix ordering (KBO)* and the *lexicographic path ordering (LPO)*. Definitions, results, and references on orderings can be found in, e.g., [4]. Since we have to deal with constraints involving finitely (but arbitrarily) many new constants, we consider a countable set[10] $\mathcal{K}$ disjoint from Σ to form the expanded signature $\Sigma^{\mathcal{K}}$. We collect all needed data in the following:

Definition 5.1 (Suitable Ordering Triple)**.** *A* suitable ordering triple *is a triple* $(\Sigma, \mathcal{K}, \succ)$ *where:* (a) Σ *is a finite signature;* (b) $\mathcal{K} := \{c_1, c_2, c_3, \dots\}$ *is a countably infinite set of constant symbols such that* Σ *and* $\mathcal{K}$ *are disjoint;* (c) $\succ$ *is a reduction ordering over* $\Sigma^{\mathcal{K}}$*-terms satisfying the following conditions:*

(i) $\succ$ *is total on ground* $\Sigma^{\mathcal{K}}$*-terms;*

(ii) *for every ground* $\Sigma^{\mathcal{K}}$*-term* t *with root symbol* $f \in \Sigma$ *and for every* $c_i \in \mathcal{K}$*, we have* $t \succ c_i$*;*

(iii) *for* $c_i, c_j \in \mathcal{K}$*, we have* $c_i \succ c_j$ *iff* $i > j$*.*

The above conditions on the reduction ordering are similar to those adopted in [2, 1] to build rewrite-based decision procedures for the constraint satisfiability problem in theories of data structures, fragments of integer arithmetic, and their combinations. It is indeed very easy and natural to produce suitable ordering triples: for instance, if an LPO is adopted, it is sufficient to take a total precedence $>_p$ satisfying the condition $f >_p c_i >_p c_j$, for $f \in \Sigma$, $c_i \in \mathcal{K}$, $c_j \in \mathcal{K}$ and $i > j$.

Another key characteristic of a rewrite-based inference system is the possibility of associating a model to the set of derived clauses, defined by building incrementally a convergent term rewriting system.

Let $(\Sigma, \mathcal{K}, \succ)$ be a suitable ordering triple and let S be a set of $\Sigma^{\mathcal{K}}$-clauses not containing the empty clause. The set $gr(S)$ contains all ground $\Sigma^{\mathcal{K}}$-clauses that are instances of clauses in S. By transfinite induction on $C \in gr(S)$, we simultaneously define $Gen(C)$ and the ground rewrite system R_C as follows:

(a) $R_C := \bigcup_{D \in gr(S), C \succ D} Gen(D)$;

(b) $Gen(C) := \{l \to r\}$ in case C is of the kind $\Delta_1 \Rightarrow l = r, \Delta_2$ and the following conditions are satisfied:

1. $R_C \not\models \Delta_1 \Rightarrow \Delta_2$, i.e. (i) for each $l = r \in \Delta_1$, l and r have the same normal form with respect to R_C (in symbols, $l \downarrow_{R_C} r$) and (ii) for each $s = t \in \Delta_2$, $s \not\downarrow_{R_C} t$;
2. $l \succ r$, $l \succ u$ (for all u occurring in Δ_1), $\{l, r\} \succ^{ms} \{u, v\}$, for every equation $u = v$ occurring in Δ_2, where $\succ^{ms}$ is the multi-set extension [4] of $\succ$;
3. l is not reducible by R_C, and
4. $R_C \not\models r = t'$, for every equation of the kind $l = t'$ occurring in Δ_2;

[10] Usual results on orderings can be extended to infinite signatures, see [15]; notice however that one can keep the signature $\Sigma^{\mathcal{K}}$ finite, by coding c_i as $s^i(0)$ (for new symbols $s, 0$), like e.g. in [9].

(c) $Gen(C) := \emptyset$, otherwise.

We say that C is productive if $Gen(C) \neq \emptyset$. Finally, let $R_S := \bigcup_{C \in gr(S)} Gen(C)$. Note that R_S is a convergent rewrite system, by conditions 2 and 3 above.

A set of clauses is *saturated* with respect to an inference system, if any clause that can be inferred from S is redundant in S (see, e.g., [7]). In a more abstract treatment, that makes saturation independent of the inference system and only requires a well-founded ordering on proofs, a set of formulæ is *saturated* if it contains all the premises of all normal-form proofs in the theory [6]. For the purposes of this paper, we are interested in a semantic notion of saturation based on model generation.

Definition 5.2. *A set S of $\Sigma^{\mathcal{K}}$-clauses is* model-saturated *iff the rewrite system R_S is a model of S (i.e. the quotient of the Herbrand universe of $\Sigma^{\mathcal{K}}$ modulo R_S-convergence is a model of the universal closures of the clauses in S).*

The following definition of reasoning module is precisely what we need to prove the main technical Lemma 5.9 below.

Definition 5.3 (Superposition Module)**.** *Let $(\Sigma, \mathcal{K}, \succ)$ be a suitable ordering triple. A* superposition module *$\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ is a computable function which takes a finite set S_0 of $\Sigma^{\mathcal{K}}$-clauses as input and returns a (possibly infinite) sequence*

$$S_0, S_1, \ldots, S_n, \ldots \tag{3}$$

of finite sets of $\Sigma^{\mathcal{K}}$-clauses, called an S_0-derivation*, such that (*i*) if S_0 is unsatisfiable, then there exists $k \geq 0$ such that the empty clause is in S_k; (*ii*) if S_0 is satisfiable, then the set*

$$S_\infty := \bigcup_{j \geq 0} \bigcap_{i \geq j} S_i$$

of persistent clauses *is model-saturated, and (*iii*) the sets S_i and S_j are logically equivalent for $(0 \leq i, j \leq \infty)$. We say that $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$* terminates *on the set of $\Sigma^{\mathcal{K}}$-clauses S_0 iff the S_0-derivation* (3) *is finite.*

Superposition modules are *deterministic*, i.e. there exists just one S_0-derivation starting with a given finite set S_0 of clauses. Any implementation of the superposition calculus [18] together with a fair strategy satisfies Definition 5.3.

5.2 Superposition Modules and Rewrite-based Decision Procedures

For the proofs below, we need a class of superposition modules which are invariant (in a sense to be made precise) under certain renamings of finitely many constants. Formally, an *n-shifting* (where n is an integer such that $n > 0$) is the operation that applied to a $\Sigma^{\mathcal{K}}$-expression E returns the $\Sigma^{\mathcal{K}}$-expression E^{+n} obtained from E by simultaneously replacing each occurrence of the free constant $c_i \in \mathcal{K}$ by the free constant c_{i+n}, for $i > 0$ (where the word 'expression' may denote a term, a literal, a clause, or a set of clauses). In practice, an n-shifting enlarges the set of free constants occurring in the set of clauses by adding the extra constants $c_1, \ldots, c_n$ that are not in the range of the function $(\cdot)^{+n}$.

Example 5.4. *Let us consider the set* $S := \{f(c_1, c_4) = c_1, f(f(c_1, c_4), c_4) = c_2\}$ *of ground* $\Sigma^{\mathcal{K}}$*-literals where* $\Sigma := \{f\}$ *and* $\mathcal{K} := \{c_1, c_2, \dots\}$*. Then, we have that* $S^{+5} := \{f(c_6, c_9) = c_6, f(f(c_6, c_9), c_9) = c_7\}$.

Definition 5.5 (Invariant Superposition Module)**.** *Let* $(\Sigma, \mathcal{K}, \succ)$ *be a suitable ordering triple. A superposition module* $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ *is* invariant *iff for every* S_0*-derivation* $S_0, S_1, \dots, S_j, \dots$ *(with* S_0 *being a set of* $\Sigma^{\mathcal{K}}$*-clauses), we have that* $(S_0)^{+n}, (S_1)^{+n}, \dots, (S_j)^{+n}, \dots$ *is an* $(S_0)^{+n}$*-derivation, for all* $n \geq 0$.

Most of the actual implementations of superposition are *stable under signature extensions* (this is so because they need to handle Skolem symbols) and hence, the behavior of a superposition prover is not affected by any proper extension of the signature and the ordering. The property of producing derivations being invariant under shifting is weaker than stability under signature extensions. As a consequence, any superposition prover can be turned into an invariant superposition module. However, not all possible implementations of the superposition calculus are invariant superposition modules, as we shall discuss in Appendix B.

Example 5.6. *Suppose that in the suitable ordering triple* $(\Sigma, \mathcal{K}, \succ)$*, the term ordering* $\succ$ *is an LPO whose precedence satisfies* $f >_p c_i >_p c_j$ *(for* $f \in \Sigma, c_i \in \mathcal{K}, c_j \in \mathcal{K}$*,* $i > j$*). Let us consider the superposition module given by the standard superposition calculus (see Appendix B) and let us take again the situation in Example 5.4. The (model-)saturated set output by* $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ *when taking* S *as input is* $S_s := \{f(c_1, c_4) = c_1, c_2 = c_1\}$*. It is not difficult to see that the set* $(S_s)^{+5} := \{f(c_6, c_9) = c_6, c_7 = c_6\}$ *is exactly the set that we would obtain as output by the superposition module* $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ *when taking as input the set* $(S)^{+5}$ *(see Example 5.4).*

Definition 5.7. *Let* $(\Sigma, \mathcal{K}, \succ)$ *be a suitable ordering triple. A universal and finitely axiomatized* Σ*-theory* T *is* $\exists$-superposition-decidable *iff there exists an invariant superposition module* $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ *that is guaranteed to terminate when taking as input* $T \cup \Gamma$*, where* Γ *is a* $\Sigma^{\mathcal{K}}$*-constraint.*

From the termination results for superposition given in [2, 1], it follows that theories such as equality, (possibly cyclic) lists, arrays, and so on are $\exists$-decidable by superposition. According to Definition 5.7, any theory T which is $\exists$-superposition-decidable is $\exists$-decidable. In the following, we show that T is also $\exists_\infty$-decidable, which is the second main result of the paper.

5.3 Invariant Superposition Modules and Cardinality Constraints

A *variable clause* is a clause containing only equations between variables or their negations. The *antecedent-mgu* (a-mgu, for short) of a variable clause $\Delta_1 \Rightarrow \Delta_2$ is the most general unifier of the unification problem $\{x \stackrel{?}{=} y \mid x = y \in \Delta_1\}$. A *cardinality constraint clause* is a variable clause $\Delta_1 \Rightarrow \Delta_2$ such that $\Rightarrow \Delta_2\mu$ does not contain any trivial equation like $x = x$, where μ is the a-mgu of $\Delta_1 \Rightarrow \Delta_2$; the number of free variables of $\Delta_2\mu$ is called the *cardinal* of the cardinality constraint clause $\Delta_1 \Rightarrow \Delta_2$. For example, the clause $x = y \Rightarrow y = z_1, x = z_2$ is a cardinality constraint clause whose cardinal is 3 (notice that this clause is true only in the one-element model).

Lemma 5.8. *If a satisfiable set* S *of clauses contains a cardinality constraint clause* $\Delta_1 \Rightarrow \Delta_2$*, then* S *cannot have a model whose domain is larger than the cardinal of* $\Delta_1 \Rightarrow \Delta_2$.

Proof. Let μ be the a-mgu of $\Delta_1 \Rightarrow \Delta_2$. By definition of a cardinality constraint clause, the clause $\Rightarrow \Delta_2\mu$ does not contain trivial equations; if n is the number of distinct variables in $\Rightarrow \Delta_2\mu$, then there cannot be more than $n-1$ distinct elements in any model of S. □

The next crucial lemma expresses the property that an invariant superposition module will discover a cardinality constraint clause whenever the input set of clauses does not admit infinite models. In Appendix B, we illustrate this behaviour by showing how the superposition calculus can derive a cardinality constraint clause from $\Rightarrow x = a, x = b$.

Lemma 5.9. *Let $(\Sigma, \mathcal{K}, \succ)$ be a suitable ordering triple. Let $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ be an invariant superposition module. If S_0 is a satisfiable finite set of clauses, then the following conditions are equivalent:*

(i) *the set S_∞ of persistent clauses in an S_0-derivation of $\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ contains a cardinality constraint clause;*

(ii) *S_0 does not admit infinite models.*

Proof. The implication (i) ⇒ (ii) is proved by Lemma 5.8. To show (ii) ⇒ (i), assume that the set S_0 does not have a model whose domain is infinite. By Lemma 3.3, there must exist a natural number N such that every model $\mathcal{M}$ of S_0 has a domain with at most N elements. Since a cardinality constraint clause does not contain constants, it is in S_∞ iff it is in $(S_\infty)^{+N}$. Hence, by Definition 5.5 of an invariant superposition module (considering $(S_0)^{+N}$ rather than S_0, if needed) we are free to assume that the constants $\{c_1, \ldots, c_N\}$ do not occur in S_∞. Recall also that, according to the definition of a suitable ordering triple, the constants $\{c_1, \ldots, c_N\}$ are the smallest ground $\Sigma^{\mathcal{K}}$-terms.

According to the definition of superposition module (cf. Definition 5.3), since S_0 is assumed to be satisfiable, S_∞ is model-saturated, which means that the convergent rewrite system R_{S_∞} is a model of S_∞ (hence also of S_0, which is logically equivalent to S_∞). Now, since S_0 does not have a model whose domain is of cardinality N or greater, there is at least one constant among $c_1, \ldots, c_N$ which is not in normal form (with respect to R_{S_∞}). Assume that c_i is not in normal form (with respect to R_{S_∞}) and that each c_j (for $j < i$) is. By model generation (see section 5.1), to reduce c_i we need a rule $l \to r$ from a productive clause C of the kind $\Delta_1 \Rightarrow l = r, \Delta_2 \in gr(S_\infty)$; furthermore, c_i can be reduced only to c_j for $j < i$. The maximality condition 2 of model generation in Section 5.1 on l implies that l is c_i and that the remaining terms in C are of the kind c_j for $j \leq i$.[11] By condition 1 of model generation in Section 5.1, the fact that all terms c_j $(j < i)$ are in R_{S_∞}-normal form, and the fact that R_{S_∞} is a convergent rewrite system extending R_C, it follows that each equation in Δ_1 is of the form $c_j = c_j$. Furthermore, again by condition 1 of model generation in Section 5.1, there is no (trivial) equality of the form $c_j = c_j$ in Δ_2. Since the constants $\{c_1, \ldots, c_N\}$ do not occur in S_∞, we are entitled to conclude that the productive clause $\Delta_1 \Rightarrow l = r, \Delta_2$ is the ground instance of a variable clause, i.e. there must exist a variable clause $\tilde{C}$ of the form $\tilde{\Delta}_1 \Rightarrow \tilde{l} = \tilde{r}, \tilde{\Delta}_2$ in S_∞ such that $\tilde{C}\theta \equiv C$ for some ground substitution θ. Since the

[11] More precisely (this is important for the proof): terms occurring positively can only be c_j for $j \leq i$ and terms occurring negatively can only be c_j for $j < i$.

antecedent of C consists of trivial equalities, θ is less general than μ, where μ is the a-mgu of $\tilde{C}$, i.e. we have that $\theta = \mu\theta'$ for some substitution θ'. Furthermore, since there are no positive trivial equalities in $C \equiv \tilde{C}\mu\theta'$, there are no positive trivial equalities in $\tilde{C}\mu$ either, which implies that $\tilde{C}$ is a cardinality constraint clause belonging to S_∞. □

The following result immediately follows from Lemma 5.9 above, because unsatisfiability in infinite models can be detected by looking for a cardinality constraint clause among the finitely many final clauses of a terminating derivation:

Theorem 5.10. *Let T be a finitely axiomatized universal Σ-theory where Σ is finite. If T is $\exists$-superposition-decidable, then T is strongly $\exists_\infty$-decidable.*

5.4 Combining Superposition Modules and SMT Procedures

Invariant superposition modules provide us with means to check whether a theory is strongly decidable or not (and this answers *QUESTION 2* in Section 3.2). However, the situation is not really clear in practice. By using available state-of-the-art implementations of the superposition calculus, such as SPASS [28] or E [24], with suitable ordering, we have run concrete invariant superposition modules for a theory $T^{\leq k}$, admitting only finite models with at most $k-1$ elements, axiomatized by an appropriate "at most" cardinality constraint, see (2). Indeed, according to Definition 5.5, the hard part is to prove termination for arbitrary input clauses of the form $T^{\leq k} \cup \Gamma$, where Γ is a set of ground literals. Our preliminary experiments were quite discouraging. In fact, both SPASS and E were able to handle only the trivial theory $T^{\leq 1}$ (axiomatized by $\Rightarrow x = y$). Already for $T^{\leq 2}$ (axiomatized by $\Rightarrow x = y, x = z, y = z$), the provers do not terminate in a reasonable amount of time although we experimented with various settings. For example, while SPASS is capable of finding a saturation for $T^{\leq 2} \cup \Gamma$ when $\Gamma := \emptyset$, it seems to diverge when $\Gamma := \{a \neq b\}$. This seems to dramatically reduce the scope of applicability of Theorem 5.10 and hence of Theorem 3.6.

Fortunately, this problem can be solved by the following two observations. First, although a superposition module may not terminate on instances of the constraint satisfiability problem of the form $T \cup \Gamma$, where Γ is a constraint and T does not admit infinite models (such as $T^{\leq k}$, above), Lemma 5.9 ensures that a cardinality constraint clause will eventually be derived in a finite amount of time: if a clause C is in the set S_∞ of persistent clauses of a derivation $S_0, S_1, \ldots$, then there must exists an integer $k \geq 0$ such that $C \in S_k$ (recall Definition 5.3). Second, when a cardinality constraint clause C is derived from $T \cup \Gamma$, a bound on the cardinality of the domains of any model can be immediately obtained by the cardinal associated to C. It is possible to use such a bound to build an equisatisfiable set of clauses (see Figure 1) and pass it to an efficient decision procedure for the pure theory of equality, based on congruence closure, such as those provided by many SMT tools (see, e.g., [10, 3, 11, 12]). The observations above motivate the following relaxation of the notion of an $\exists$-superposition-decidable theory.

Definition 5.11. *Let $(\Sigma, \mathcal{K}, \succ)$ be a suitable ordering triple. A universal and finitely axiomatized Σ-theory T is* weakly-$\exists$-superposition-decidable *iff there exists an invariant superposition module*

function *Grounding* (N : *integer*, T: *axioms*, Γ: *Ground literals*)

1 introduce fresh constants $c_1, \ldots, c_N$;

2 for every k-ary function symbol f in $\Gamma \cup T$ (with $k \geq 0$), generate the positive clauses

$$\bigvee_{i=1}^{N} f(a_1, \ldots, a_k) = c_i$$

for every $a_1, \ldots, a_k \in \{c_1, \ldots, c_N\}$ and let E be the resulting set of clauses;

3 for every clause $C \in T$, instantiate C in all possible ways by ground substitutions whose range is the set $\{c_1, \ldots, c_N\}$ and let T_g be the resulting set of clauses;

4 return the set $T_g \cup E \cup \Gamma$.

end

Figure 1: Computing equisatisfiable sets of ground clauses for instances of the constraint satisfiability problem of theories with no infinite models

$\mathcal{SP}(\Sigma, \mathcal{K}, \succ)$ *such that for every* $\Sigma^{\mathcal{K}}$*-constraint* Γ*, any* $T \cup \Gamma$*-derivation either* (i) *terminates or* (ii) *generates a cardinality constraint clause.*

We can easily adapt Theorem 5.10 to this new definition.

Theorem 5.12. *Let* T *be a universal and finitely axiomatized* Σ*-theory, where* Σ *is finite. If* T *is weakly-*$\exists$*-superposition-decidable, then* T *is strongly* $\exists_\infty$*-decidable.*

Proof. Decidability of Σ-constraints in models of T can be obtained by halting the invariant superposition module and then using any SMT procedure for the theory of equality with the set of clauses obtained by the function *Grounding* of Figure 1. Decidability in infinite models is answered negatively if a cardinality constraint clause is generated; otherwise, we have termination of the invariant superposition module and if the empty clause is not produced, satisfiability is reported by Lemma 5.9. □

6 Conclusion and Future Work

By classifying the component theories according to the decidability of constraint satisfiability problems in finite and infinite models, respectively, we exhibited a theory T_1 such that T_1-satisfiability is decidable, but T_1-satisfiability in infinite models is undecidable. It follows that satisfiability in $T_1 \cup T_2$ is undecidable, whenever T_2 has only infinite models, even if signatures are disjoint and satisfiability in T_2 is decidable. In the second part of the paper we strengthened the Nelson-Oppen combination result, by showing that it applies to theories over disjoint signatures, whose satisfiability problem, in either finite or infinite models, is decidable. We showed that this result covers decision procedures based on superposition, generalizing the recent approach of [1].

An interesting line of future work consists of finding *ad hoc* simplification rules which allow the superposition calculus to terminate on theories that do not admit infinite models such as the $T^{\leq k}$'s considered in Section 5.4.

References

[1] Alessandro Armando, Maria Paola Bonacina, Silvio Ranise, and Stephan Schulz. On a rewriting approach to satisfiability procedures: extension, combination of theories and an experimental appraisal. In *Proc. of the 5th Int. Workshop on Frontiers of Combining Systems (FroCoS'05)*, volume 3717 of *LNCS*, pages 65–80. Springer, 2005.

[2] Alessandro Armando, Silvio Ranise, and Michaël Rusinowitch. A rewriting approach to satisfiability procedures. *Information and Computation*, 183(2):140–164, 2003. RTA 2001 (Utrecht).

[3] Gilles Audemard, Piergiorgio Bertoli, Alessandro Cimatti, Artur Korniłowicz, and Roberto Sebastiani. A SAT based approach for solving formulas over boolean and linear mathematical propositions. In *Proc. International Conference on Automated Deduction (CADE-18)*, volume 2392 of *LNCS*, pages 195–210. Springer, 2002.

[4] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, United Kingdom, 1998.

[5] Franz Baader and Cesare Tinelli. Deciding the word problem in the union of equational theories. *Information and Computation*, 178(2):346–390, December 2002.

[6] Maria Paola Bonacina and Nachum Dershowitz. Abstract canonical inference. *ACM Transactions on Computational Logic*, (to appear), 2006.

[7] Maria Paola Bonacina and Jieh Hsiang. Towards a foundation of completion procedures as semidecision procedures. *Theoretical Computer Science*, 146:199–242, July 1995.

[8] Hubert Comon. Solving symbolic ordering constraints. *International Journal of Foundations of Computer Science*, 1(4):387–412, 1990.

[9] Hubert Comon, Paliath Narendran, Robert Nieuwenhuis, and Michaël Rusinowitch. Decision problems in ordered rewriting. In *Proc. 13th IEEE Symp. Logic in Computer Science (LICS'98)*, pages 276–286, Indianapolis, Indiana, USA, 1998. IEEE Computer Society Press.

[10] David Déharbe and Silvio Ranise. Light-weight theorem proving for debugging and verifying units of code. In *Proc. of the International Conference on Software Engineering and Formal Methods (SEFM03)*, Brisbane, Australia, September 2003. IEEE Computer Society Press.

[11] Jean-Christophe Filliâtre, Sam Owre, Harald Rueß, and Natarajan Shankar. ICS: Integrated canonizer and solver. In *Proc. International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *LNCS*, pages 246–249. Springer, 2001.

[12] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. DPLL(T): Fast decision procedures. In R. Alur and D. Peled, editors, *Proc. International Conference on Computer Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pages 175–188. Springer, 2004.

[13] Silvio Ghilardi. Model theoretic methods in combined constraint satisfiability. *Journal of Automated Reasoning*, 33(3-3):221–249, 2005.

[14] Konstantin Korovin and Andrei Voronkov. Knuth-bendix constraint solving is NP-complete. *ACM Transactions on Computational Logic*, 6(2):361–388, 2005.

[15] Aart Middeldorp and Hans Zantema. Simple termination revisited. In *Proc. International Conference on Automated Deduction (CADE'94)*, LNCS, pages 451–465, Nancy, France, 1994. Springer.

[16] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. on Programming Languages and Systems*, 1(2):245–257, October 1979.

[17] Robert Nieuwenhuis and José Miguel Rivero. Practical algorithms for deciding path ordering constraint satisfaction. *Information and Computation*, 178(2):422–440, 2002.

[18] Robert Nieuwenhuis and Albert Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier and MIT Press.

[19] Piergiorgio Odifreddi. *Classical recursion theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1989. The theory of functions and sets of natural numbers, With a foreword by G. E. Sacks.

[20] Derek C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12:291–302, 1980.

[21] Don Pigozzi. The join of equational theories. *Colloquium Mathematicum*, 30(1):15–25, 1974.

[22] Mojzesz Presburger. Ueber die Vollstaendigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du I congrés de Mathématiciens des Pays Slaves*, pages 92–101, 1929.

[23] Silvio Ranise, Christophe Ringeissen, and Duc-Khanh Tran. Nelson-Oppen, Shostak and the extended canonizer: A family picture with a newborn. In Keijiro Araki and Zhiming Liu, editors, *First International Colloquium on Theoretical Aspects of Computing - ICTAC 2004*, LNCS, Guiyang, Chine, September 2004. Springer.

[24] Stephan Schulz. E - a brainiac theorem prover. *AI Communications*, 15(2/3):111–126, 2002.

[25] Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems: Proceedings of the 1st International Workshop (Munich, Germany)*, Applied Logic, pages 103–120. Kluwer Academic Publishers, March 1996.

[26] Cesare Tinelli and Calogero G. Zarba. Combining non-stably infinite theories. *Journal of Automated Reasoning*, 2006. (to appear).

[27] Dirk van Dalen. *Logic and Structure*. Springer-Verlag, 1989. Second edition.

[28] Christoph Weidenbach. Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. 2001.

A Refined Undecidability Results

Here we refine Proposition 4.1 by avoiding the use of an infinite signature like Σ_{TM_∞}.

A.1 A Variant of Theory TM_∞: TM_ω

Consider the signature Σ_{TM_ω} consisting of a constant symbol 0, a unary predicate symbol P, and two binary predicate symbols $<$ and S. The axioms of the theory TM_ω are the universal closures of the following formulæ:

$$\neg x < x \tag{4}$$

$$x < y \wedge y < z \rightarrow x < z \tag{5}$$

$$x < y \vee x = y \vee y < x \tag{6}$$

$$0 = x \vee 0 < x \tag{7}$$

$$S(x, y) \leftrightarrow (x < y \wedge \neg\exists z(x < z \wedge z < y)) \tag{8}$$

$$x < y \rightarrow \exists z(S(x, z) \wedge (z < y \vee z = y)) \tag{9}$$

$$P(x_a) \wedge S(0, x_1) \wedge \cdots \wedge S(x_{a-1}, x_a) \wedge S(x_a, x_{a+1}) \wedge \cdots \wedge S(x_{a+m-1}, x_{a+m}) \rightarrow \bot, \quad \text{if } a = \langle e, n\rangle \text{ and } k(e, n) < m \tag{10}$$

$$P(x) \wedge P(y) \rightarrow x = y \tag{11}$$

where $\langle \cdot, \cdot \rangle$ is a primitive recursive coding for pairs, i.e. a computable bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$ (we are guaranteed that the primitive recursive coding function $\langle \cdot, \cdot \rangle$ exists because of basic results about primitive recursive functions, see again [19] for details).

Two remarks are in order. First, because of axioms (4)-(9), any model of TM_ω is a Σ_{TM_ω}-structure endowed with a strict linear order with first element; moreover, every element (except the last one, if any) has an immediate successor. Second, finite models of TM_ω are initial segments of $\mathbb{N}$, whereas infinite models admit $\mathbb{N}$ as an initial segment. It is also worth to consider the last two axioms of TM_ω:

- axiom (10) means that, given a Turing Machine e, its input n, and the coding h of the pair (e, n), the atom $P(h)$ can be satisfied only in models of cardinality at most $h + k(e, n) + 1$;
- axiom (11) states that there is no model satisfying two atoms $P(a)$ and $P(b)$ if $a \neq b$. This axiom simplifies the technical development below.

Proposition A.1. *The theory TM_ω is $\exists$-decidable but it is not $\exists_\infty$-decidable.*

Proof. Let Γ be a constraint over the signature $\Sigma^{\underline{c}}$. We define a *TM_ω-guessing* $\mathcal{G}$ on Γ as a finite set of ground $\Sigma^{\underline{c}}$–literals such that (i) $\Gamma \subseteq \mathcal{G}$ and (ii) for every pair of distinct constants $a, b \in \underline{c} \cup \{0\}$, either $a < b \in \mathcal{G}$, $b < a \in \mathcal{G}$, or $a = b \in \mathcal{G}$. Clearly, Γ is TM_ω-satisfiable iff some TM_ω-guessing $\mathcal{G}$ on Γ is TM_ω-satisfiable. As a consequence, we consider the problem of deciding the satisfiability of a TM_ω-guessing $\mathcal{G}$.

Given such a TM_ω-guessing $\mathcal{G}$, notice that for $\mathcal{G}$ to be consistent, the equations belonging to $\mathcal{G}$ must induce an equivalence relation on the constants occurring in it. Let us pick a representative

constant for each equivalence class (with 0 being the representative for its class). Furthermore, let us replace all terms in $\mathcal{G}$ with the representative constants of their equivalence classes. After this transformation, without loss of generality, we can delete all equalities and inequalities from $\mathcal{G}$. Let us denote the result of such transformations still with $\mathcal{G}$. For each negative literal $\neg S(c_1, c_2)$ in $\mathcal{G}$ such that $c_1 < c_2 \in \mathcal{G}$ and $\{c_1 < a, a < c_2\} \not\subseteq \mathcal{G}$ for some constant a, we add $c_1 < c_3$, $c_3 < c_2$ to $\mathcal{G}$, where c_3 is a fresh constant. After this step, the literals of the form $a < b$ that are in $\mathcal{G}$ should put the constants in $\mathcal{G}$ in a linear order, i.e.

$$c_0 < c_1 < c_2 < \cdots < c_{s-1} < c_s.$$

Here c_0 is 0, c_i and c_j are distinct for $i \neq j$, and only inequalities of the form $c_i < c_j$ $(0 \leq i < j \leq s)$ are in $\mathcal{G}$ (if it is not so, it is because $\mathcal{G}$ contains inconsistencies from the point of view of the theory of strict linear orders with first element). Furthermore, if $S(c_i, c_j) \in \mathcal{G}$, then $j = i + 1$; otherwise, $\mathcal{G}$ is inconsistent. Thus, all literals in $\mathcal{G}$ (not containing P) are satisfied, for instance, in the linearly ordered structure containing s elements. Clearly, $\mathcal{G}$ is inconsistent if it contains a pair of complementary literals, so we suppose this is not the case. Because of axiom (11), it can contain at most one positive literal involving the predicate P and, at this point, $\mathcal{G}$ can be unsatisfiable only because of the presence of such a literal. Let this literal be $P(c_a)$; for $m = s - a$, the following inequalities

$$0 < c_1 < c_2 < \cdots < c_{a-1} < c_a < c_{a+1} < \cdots < c_{a+m}$$

are in $\mathcal{G}$. If there is $j < a$ such that $S(c_j, c_{j+1}) \notin \mathcal{G}$, then $\mathcal{G}$ is satisfiable. To see this, consider a non standard model of Arithmetic and interpret $c_{j+1}, \ldots, c_{a+m}$ as elements greater than all the standard natural numbers: if the predicate P is interpreted as the singleton subset formed by (the interpretation of) c_a, axiom (10) is true because the a-th successor of 0 is not in P. On the other hand, if $\{S(0, c_1), S(c_1, c_2), \ldots, S(c_{a-1}, c_a)\} \subseteq \mathcal{G}$, then $\mathcal{G}$ is satisfiable iff $m \leq k(e, n)$ where $a = \langle e, n \rangle$. Since $\langle e, n \rangle$ and the relation $m \leq k(e, n)$ are computable, we have a decision procedure for the constraint satisfiability problem in TM_ω.

To see that TM_ω is not $\exists_\infty$-decidable, notice that the TM_ω-constraint

$$\{S(0, c_1), S(c_1, c_2), \ldots, S(c_{a-1}, c_a), P(c_a)\},$$

(for $a = \langle e, n \rangle$) is TM_ω-satisfiable in an infinite structure iff the computation of the Turing Machine e over the input n diverges, which is obviously undecidable. □

A.2 A Variant of Theory TM_ω: $TM_{\forall\omega}$

Theory TM_ω is not universal. However, it is not difficult to find an alternative axiomatization over a finite signature $\Sigma_{TM_{\forall\omega}}$ so to define a universal theory $TM_{\forall\omega}$ which is $\exists$-decidable but it is not $\exists_\infty$-decidable. With this theory in mind, the full claim of Theorem 4.2 is proved: now the $\exists$-decidable component theories leading to undecidable combined problems are universal and signatures are always finite and disjoint (the theory of acyclic lists [20] is universal, has only infinite models and its signature is finite, so it satisfies all needed requirements for the undecidable combination with $TM_{\forall\omega}$).

The main ideas used in the definition of $TM_{\forall\omega}$ are the following: (a) we replace the binary predicate symbol S of TM_ω with a unary function symbol s; (b) we re-use axioms (4)-(7) and (c) we introduce new axioms to constrain the unary symbol s to be such that $s(x) = x$ holds iff the order $<$ has a last element which is precisely x.

In more detail, the signature of the theory $TM_{\forall\omega}$ coincides with the signature of the theory TM_ω with the exception that the binary predicate symbol S is replaced by the unary function symbol s. The axioms for $TM_{\forall\omega}$ are divided into three groups. In the first group we have axioms (4)-(7) and in the second group the following ones:

$$x = s(x) \vee x < s(x) \tag{12}$$

$$\neg(x < y \wedge y < s(x)) \tag{13}$$

$$x < y \rightarrow s(x) < y \vee s(x) = y \tag{14}$$

$$s(x) = x \wedge x < y \rightarrow \bot \tag{15}$$

Axioms (12)-(15), together with (4)-(7), state that the function s behaves like a successor function with the exception that fixed points of s are allowed (see (12)). Axiom (15) however says that the only possible fixed point of the function s is the maximum element with respect to the order $<$.

In addition to the axioms of the first two groups (namely (4)-(7) and (12)-(15)), in the third group, we have axiom (11) and the following one (which replaces (10)):

$$P(s^a(0)) \wedge s^{a+m-1}(0) < s^{a+m}(0) \rightarrow \bot \qquad \text{if } a = \langle e, n\rangle \text{ and } k(e, n) < m \tag{16}$$

Proposition A.2. *The theory $TM_{\forall\omega}$ is $\exists$-decidable but it is not $\exists_\infty$-decidable.*

Proof. The argument is similar to the argument used in the proof of Proposition A.1, with the proviso that the constraint Γ should be flattened. Moreover, once the linear order

$$c_0 < c_1 < \cdots < c_{s-1} < c_s$$

is obtained, we notice that if the literal $c_j = s(c_i)$ belongs to the guessing $\mathcal{G}$, then this is inconsistent if $j \neq i+1$ or if $j = i \neq s$. The other steps in the proof of Proposition A.1 remain unchanged. □

It is still an open problem to find an $\exists$-decidable, non $\exists_\infty$-decidable theory (in a finite signature), which is universal and *finitely axiomatized*.

B Deriving a Cardinality Constraint Clause in Practice

In this Appendix we give an example showing the content of Lemma 5.9 and we further discuss invariance as stated in Definition 5.5. Figure 2 shows the expansion inference rules of the superposition calculus used in [2, 1].

This calculus is refutationally complete: model generation technique is the main tool to show this result. However, the completeness proof in [18] makes clear that the calculus is complete as well if the ordering constraints are interpreted as *symbolic constraint solving problems* (see, e.g.,

Superposition	$\dfrac{C \vee l[u'] = r \quad D \vee u = t}{(C \vee D \vee l[t] = r)\sigma}$	$(i), \quad (ii), \quad (iii), \quad (iv)$
Paramodulation	$\dfrac{C \vee l[u'] \neq r \quad D \vee u = t}{(C \vee D \vee l[t] \neq r)\sigma}$	$(i), \quad (ii), \quad (iii), \quad (iv)$
Reflection	$\dfrac{C \vee u' \neq u}{C\sigma}$	$\forall L \in C : (u' = u)\sigma \not\prec L\sigma$
Equational Factoring	$\dfrac{C \vee u = t, u' = t'}{(C \vee t \neq t' \vee u = t')\sigma}$	$(i), \quad \forall L \in \{u' = t'\} \cup C : (u = t)\sigma \not\prec L\sigma$

where the notation $l[u']$ means that u' appears as a sub-term in l, σ is the most general unifier (mgu) of u and u', u' is not a variable in *Superposition* and *Paramodulation*, and the following abbreviations hold:

(i) is $u\sigma \not\preceq t\sigma$,

(ii) is $\forall L \in D : (u = t)\sigma \not\preceq L\sigma$,

(iii) is $l[u']\sigma \not\preceq r\sigma$, and

(iv) is $\forall L \in C : (l[u'] \bowtie r)\sigma \not\preceq L\sigma$.

Figure 2: Expansion rules: in these rules, what is below the inference line is added to the clause set that contains what is above the inference line. Premises of a rule should be renamed to have disjoint variables; $\bowtie$ is either $=$ or $\neq$, and identity is symmetrized (meaning that $s = t$ may also denote $t = s$).

[8, 17, 14]): this means that e.g. the condition (i) can be rephrased as 'there exists a ground substitution θ such that $u\sigma\theta \succ t\sigma\theta$' (where $\not\preceq$ can be replaced to $\succ$, because the ordering is total on ground terms). We can further restrict the ground substitution θ to take values in the *actual* signature (and not in a signature extending the actual one). These choices are not very convenient from a practical point of view, because the benefit of blocking some inference does not compensate the increase in complexity due to the intractability of symbolic constraint solving problems (which usually are NP-complete problems). What we want to point out here is that this interpretation of ordering constraints as symbolic constraint solving problems in the actual signature destroys invariance in the sense of Definition 5.5 and also invalidates the statement of Lemma 5.9. To see why this is the case, let $c_1 \in \mathcal{K}$ be the smallest constant in the given suitable ordering triple. A clause like $x = c_1$ can be superposed with itself if the maximality constraint is interpreted as $x \not\succeq c_1$ (and the result of the superposition is $x = y$). On the other hand, if the maximality constraint is interpreted as a symbolic constraint solving problem in the signature $\Sigma^{\mathcal{K}}$, then no superposition applies because there is no ground term smaller than c_1 in $\Sigma^{\mathcal{K}}$. Unfortunately, if we apply a +2-shifting, then the symbolic constraint $c_3 \succ x$? has, e.g., the solution $x \mapsto c_1$ and superposition is not blocked anymore. Notice also that the singleton set of clauses $\{x = c_1\}$ is model-saturated,[12] has no infinite models, but does not contain a cardinality constraint clause.

To illustrate the content of Lemma 5.9 in a simple but not entirely trivial case, let us consider

[12]Recall that we defined model-saturation of a set of clauses in terms of the rewrite system associated to the model generation construction (and not in terms of closure - up to redundancy - with respect to the rules of the calculus).

the clause $\Rightarrow x = a, x = b$, which tells us that there are at most two elements in the domain of a model (these are the interpretations of the constants a and b). It is instructive to apply to this clause the superposition calculus (in the plain Figure 2 formulation, where ordering constraints are just $\not\trianglelefteq$-conditions). The following is a derivation of a cardinality constraint clause:

1.		$\Rightarrow$	$u = a, u = b$	
2.		$\Rightarrow$	$u = a, v = a, v = u$	[*Sup* 1.1, 1.1]
3.		$\Rightarrow$	$u = a, u = v, w = v, x = a, x = w$	[*Sup* 2.0, 2.0]
4.	$a = a$	$\Rightarrow$	$u = v, w = v, u = a, u = w$	[*Fac* 3.0, 3.3]
5.		$\Rightarrow$	$u = v, w = v, u = a, u = w$	[*Ref* 4.0]
6.		$\Rightarrow$	$u = v, w = v, u = w, x = y, z = y, x = u, x = z$	[*Sup* 5.2, 5.2]

where u, v, w, x, y, and z are variables, *Sup* abbreviates *Superposition*, *Fac* abbreviates *Factoring*, *Ref* abbreviates *Reflection*, and the sequences of non-negative integers separated by '.' denote positions. With a little bit of effort, it is possible to derive (by continuing the application of the rules of the calculus) a cardinality constraint clause whose cardinal is 3:

7.	$v = y$	$\Rightarrow$	$z = y, w = v, z = w, x = y, x = z, x = z$	[*Fac* 6.0, 6.4]
8.		$\Rightarrow$	$z = y, w = y, z = w, x = y, x = z, x = z$	[*Ref* 7.0]
9.	$y = y$	$\Rightarrow$	$z = y, x = y, z = x, x = z, x = z$	[*Fac* 8.1, 8.3]
10.		$\Rightarrow$	$z = y, x = y, z = x, x = z, x = z$	[*Ref* 9.0]
11.	$z = z$	$\Rightarrow$	$z = y, x = y, z = x, x = z$	[*Fac* 10.3, 10.4]
12.		$\Rightarrow$	$z = y, x = y, z = x, x = z$	[*Ref* 11.0]
13.	$z = z$	$\Rightarrow$	$z = y, x = y, z = x$	[*Fac* 12.2, 12.3]
14.		$\Rightarrow$	$z = y, x = y, z = x$	[*Ref* 13.0]

Cardinality constraint clauses are always derived by common superposition provers, according to Lemma 5.9, when saturating sets of clauses not admitting infinite models. Such derivations, however, even in simple cases like the one above, seems to take considerable amount of time in state-of-the-art provers.