# Variant-Based Satisfiability in Initial Algebras

José Meseguer

Department of Computer Science
University of Illinois at Urbana-Champaign
meseguer@illinois.edu

*A Fuensanta, por "La Tregua," donde surgieron estas ideas*

**Abstract.** Although different satisfiability decision procedures can be combined by algorithms such as those of Nelson-Oppen or Shostak, current tools typically can only support a finite number of theories to use in such combinations. To make SMT solving more widely applicable one needs *theory-generic* satisfiability algorithms allowing a potentially infinite number of decidable theories to be *user-definable*, instead of needing to be built in by tool implementers. This work studies how *folding variant narrowing*, a generic unification algorithm that offers good extensibility in unification theory, can be extended to a generic *variant-based satisfiability* algorithm for the initial algebras of user-specified input theories when such theories satisfy Comon and Delaune's *finite variant property* (FVP) and some extra conditions. Several, increasingly larger infinite classes of theories whose initial algebras enjoy decidable variant-based satisfiability are identified and illustrated with examples. A method based on *descent maps* to bring other theories into these classes and to improve the generic algorithm's efficiency is also proposed.
**Keywords**: finite variant property (FVP), constructor variant, constructor unifier, folding variant narrowing, satisfiability in initial algebras.

## 1 Introduction

The use of decision procedures for theories axiomatizing data structures and functions commonly occurring in software and hardware systems is currently one of the most effective methods at the heart of state-of-the-art theorem provers and model checkers. It offers the promise, and often even the reality, of scaling up such verification efforts to handle large systems used in industrial practice. In the area of decision procedures two important phases stand out. The first is the discovery in the late 70's and early 80's of *combination methods* by Nelson and Oppen [79] and Shostak [84] to achieve satisfiability in combinations of decidable theories. The second is the marriage of SAT-solving technology with decision procedures for certain theories, an approach pioneered independently by a number of different groups [5,46,7,14,78,47] and distilled in the influential

DPLL(T) architecture [81]. This approach has been key to the success of SMT, as witnessed by a vast literature on the subject.

However, one important challenge is the lack of *extensibility* of current SMT tools. This may seem somewhat paradoxical to say, since obviously the Nelson-Oppen (NO) combination method [79,82] offers unlimited extensibility by theory combinations under some conditions on the combined theories. This is true enough, but:

1. One needs to have algorithms and implementations for each of the theories supported by the SMT solver, which requires a non-trivial effort and in any case limits at any given time each SMT solver to support a *finite* (and in practice not very large) library of theories that it can handle.
2. What we need are *theory-generic* —i.e., not for a single theory, but for a possibly infinite class of theories—satisfiability decision procedures, so that the input theories are easily *user-definable*. In this way, an SMT solver's repertory of individual decidable theories becomes potentially *infinite* and easily specifiable by the tool's *users*, as opposed to its implementers.

Achieving extensibility in this, more ambitious sense can have large payoffs for SMT solving technology, because it can widely extend both its *scope* and its *effectiveness*. In formal verification practice this would allow automating larger fragments of the verification effort, both in theorem proving and in model checking, and therefore scaling up to effectively handle larger problems.

This paper is all about making SMT solving extensible in the just-mentioned sense by what I call *variant-based satisfiability* methods. The best way for me to explain the key ideas is to place them in the context of a recent sea change in *unification theory* that has been quietly taking place thanks to *variant-based unification* [42,43], inspired by the Comon and Delaune notion of variant [32].

Unification theory is not just a *neighboring area* of SMT solving, but actually a *subfield*. Specifically, the subfield obtained by: (i) considering theories of the form $th(T_{\Sigma/E}(X))$, associated to equational theories $(\Sigma, E)$, where $th(T_{\Sigma/E}(X))$ denotes the theory of the free $(\Sigma, E)$-algebra $T_{\Sigma/E}(X)$ on countably many variables $X$, and (ii) restricting ourselves to *positive* quantifier-free (QF) formulas of the form $\varphi = \bigvee_i \bigwedge G_i$, with each $\bigwedge G_i$ a conjunction of equations. A finitary $E$-unification algorithm then gives us a *decision procedure* for satisfiability of such formulas $\varphi$ not only in the *free* $(\Sigma, E)$-algebra $T_{\Sigma/E}(X)$, but also in the *initial* $(\Sigma, E)$-algebra $T_{\Sigma/E}$ when all sorts of $T_{\Sigma/E}$ are non-empty.

Unification theory is not only a subfield of SMT solving but what might be called a *microcosm*, where many problems and challenges of SMT solving already show up, including the extensibility problem. For example, the Nelson-Oppen (NO) combination algorithm [79,82] is mirrored by algorithms for combining unification procedures, such as those of Baader and Schulz [8] and Boudet [20] (see [10] for a unified treatment of both NO and the Baader-Schulz algorithms). Also, as for SMT solving, extensibility *is* a problem for the exact same reasons: although combination methods exist, $E$-unification algorithms require substantial implementation efforts and a tool can only support so many of them.

One important advantage of unification theory is that it has had for a long time *generic* $E$-unification *semi-algorithms*, namely, *narrowing-based* [87,45,60,61] and *transformation-based* [48,88] ones. But one important drawback of these semi-algorithms is that, since $E$-unification for arbitrary $E$ is undecidable, in general they only provide a *semi-decision* procedure, which is useless for *deciding* unifiability, i.e., satisfiability of formulas $\varphi = \bigvee_i \bigwedge G_i$ in the initial algebra $T_{\Sigma/E}$, *unless* they can be proved *terminating* for a given equational theory $E$. For theories $E$ whose equations can be oriented as convergent rewrite rules $R$, some termination results for narrowing-based unification, mostly centered on special input theories for the *basic narrowing* strategy [60], do exist for some quite restrictive classes of rules $R$ (see [1,2], and references there, for a comprehensive and up-to-date treatment). Instead, the more general case of termination for narrowing-based unification for equational theories $E \uplus B$ for which the equations $E$ can be oriented as convergent rules $R$ *modulo* axioms $B$ having a finitary $B$-unification algorithm, has been a real *terra incognita* until very recently, because *negative* results, like the impossibility of using basic narrowing when $B$ is a set of associative-commutative (AC) axioms [32], seemed to dash any hopes not just of termination, but even of efficient implementation. Many of these limitations have now disappeared thanks to the *folding variant narrowing* algorithm [42,43]. Let me summarize the current state of the art after [43]:

1. When $B$ has a finitary unification algorithm, folding variant narrowing with convergent oriented equations $E$ modulo $B$ will terminate on any input term (including unification problems expressed in an extended signature) iff $E \uplus B$ has the *finite variant property*[1] (FVP) in the Comon and Delaune sense [32].
2. No other complete narrowing strategy can terminate more often than folding variant narrowing; in particular, basic narrowing (when applicable, e.g., $B = \varnothing$) terminates strictly less often.
3. FVP is a semi-decidable property, and when it actually holds can be easily checked by existing tools, assuming convergence [24].
4. Both folding variant narrowing and variant-based unification for theories $E \uplus B$, where $B$ can be any combination of associativity, commutativity and identity axioms are well supported by Maude [26] in its latest 2.7.1 version.

There are by now papers, e.g., [32,41,40], many cryptographic protocol specifications, e.g., [41,92,55,23,83], and several verification tools, e.g., [41,23,83], demonstrating that FVP equational theories are omni-present in cryptographic protocol verification and that variant-based unification and narrowing are very general and effective formal reasoning methods to verify such protocols. In this paper I give many examples showing that, in a similar way, QF satisfiability in initial algebras of FVP theories is decidable under reasonable conditions.

[1] Roughly (see Def. 5 for a more precise definition), $u$ is an $E, B$-*variant* of a term $t$ if $u$ is the $E, B$-canonical form of a substitution instance, $t\theta$, of $t$. Therefore, the variants of $t$ are intuitively the "irreducible patterns" to which $t$ can be symbolically evaluated by the rules $E$ modulo $B$. $E \uplus B$ has the *finite variant property* if there is a *finite* set of *most general* variants, which are computed by *folding variant narrowing*.

The key question addressed in this paper should now be obvious: can the good properties of variant-based unification as a *theory-generic*, finitary $E \uplus B$-unification algorithm for FVP theories be extended to a, likewise theory-generic, variant-based $E \uplus B$-satisfiability algorithm for the initial algebras $T_{\Sigma/E \uplus B}$ of an infinite number of such FVP theories $E \uplus B$ under suitable conditions? If this were possible, the advances in increasing the *extensibility* of unification theory could then be leveraged to make SMT solving substantially more extensible than it is at present. Answering this question is non-trivial, because unification only deals with positive, i.e., negation-free, formulas, whereas satisfiability must deal with *all* QF formulas. This is precisely what is done in this work, which answers this main question in the affirmative as follows:

1. After some preliminaries in Section 2, Section 3 discusses an incorrect first attempt, in [32], to relate satisfiability and initial FVP algebras. Section 4 then proposes new notions of *constructor variant* and *constructor unifier* as key concepts towards a solution.
2. Section 5 gives a general "descent theorem" reducing satisfiability in an initial algebra to satisfiability in a simpler initial algebra on a subsignature $\Omega$ of constructors, and outlines a general satisfiability algorithm when the initial algebra of constructors has decidable satisfiability for QF formulas.
3. General conditions under which the initial algebra of constructors associated to an initial algebra $T_{\Sigma/E \uplus B}$ has decidable satisfiability and makes, in turn, satisfiability in $T_{\Sigma/E \uplus B}$ decidable are investigated. A key notion is that of an *OS-compact theory*, which generalizes in several ways that of a compact theory in [30]. In particular, it is shown that $T_{\Omega/B}$ has decidable QF satisfiability for $B$ any combination of associativity, commutativity and identity axioms, except associativity without commutativity; furthermore, various relevant examples of decidable initial algebras whose initial algebra of constructors are of the form $T_{\Omega/B}$ are given.
4. Section 7 shows that various parameterized data types, such as lists, compact lists [35,34], multisets, and hereditarily finite (HF) sets, are *satisfiability-preserving* under very general conditions; that is, they map a target initial algebra with decidable QF satisfiability, like integers with addition, to the initial algebra of the corresponding instance of the parameterized module, like sets of integers, also with decidable QF satisfiability.
5. Section 8 then brings all the notions in Sections 5–7 under the common notion of a *descent map* relating a more complex theory to a simpler one. Descent maps can be used to: (i) specify and prove satisfiability algorithms in a modular way, and prove satisfiability in cases where the initial algebra of constructors of a given FVP initial algebra $T_{\Sigma/E \uplus B}$ is not OS-compact; and (ii) substantially reduce the computational cost of satisfiability algorithms by mapping a theory to a simpler *core theory* whose initial algebra is satisfiable.
6. Related work is discussed in Section 9; and a fuller discussion of the entire work and its future directions is given in Section 10.

This paper is an extended and improved version of the conference paper [72]. In comparison with [72], the following are new contributions:

1. Proofs of all results are included (no proofs were given in [72]).
2. Section 4 has been substantially expanded and improved.
3. A much fuller discussion of satisfiability in parameterized data types is given in Section 7 (only HF sets were discussed in [72]).
4. Section 8 is entirely new.
5. A fuller collection of examples is given (23, versus 12 in [72]).

## 2 Order-Sorted Algebra, Rewriting, and Variants

I summarize the order-sorted algebra, order-sorted rewriting, and FVP notions needed in the paper. The material, adapted from [73,43], extends ideas in [54,32]. It assumes the notions of many-sorted signature and many-sorted algebra, e.g., [38], which include unsorted signatures and algebras as a special case.

**Definition 1.** *An* order-sorted (OS) signature *is a triple* $\Sigma = ((S, \leqslant), \Sigma)$ *with* $(S, \leqslant)$ *a poset and* $(S, \Sigma)$ *a many-sorted signature.* $\widehat{S} = S/\equiv_{\leqslant}$*, the quotient of* $S$ *under the equivalence relation* $\equiv_{\leqslant} = (\leqslant \cup \geqslant)^+$*, is called the set of* connected components *of* $(S, \leqslant)$*. The order* $\leqslant$ *and equivalence* $\equiv_{\leqslant}$ *are extended to sequences of same length in the usual way, e.g.,* $s'_1 \ldots s'_n \leqslant s_1 \ldots s_n$ *iff* $s'_i \leqslant s_i$*,* $1 \leqslant i \leqslant n$*.* $\Sigma$ *is called* sensible *if for any two* $f : w \to s, f : w' \to s' \in \Sigma$*, with* $w$ *and* $w'$ *of same length, we have* $w \equiv_{\leqslant} w' \;\Rightarrow\; s \equiv_{\leqslant} s'$*.* $A$ many-sorted signature $\Sigma$ *is the special case where the poset* $(S, \leqslant)$ *is discrete, i.e.,* $s \leqslant s'$ *iff* $s = s'$*.* $\Sigma = ((S, \leqslant), \Sigma)$ *is a* subsignature *of* $\Sigma' = ((S', \leqslant'), \Sigma')$*, denoted* $\Sigma \subseteq \Sigma'$*, iff* $S \subseteq S'$*,* $\leqslant \subseteq \leqslant'$*, and* $\Sigma \subseteq \Sigma'$*.*

*For connected components* $[s_1], \ldots, [s_n], [s] \in \widehat{S}$

$$f^{[s_1]\ldots[s_n]}_{[s]} = \{f : s'_1 \ldots s'_n \to s' \in \Sigma \mid s'_i \in [s_i],\;\; 1 \leqslant i \leqslant n,\;\; s' \in [s]\}$$

*denotes the family of "subsort polymorphic" operators* $f$*.* □

I will always assume that $\Sigma$'s poset of sorts $(S, \leqslant)$ is *locally finite*, that is, that for any $s \in S$ its connected component $[s]$ is a finite set.

**Definition 2.** *For* $\Sigma = (S, \leqslant, \Sigma)$ *an OS signature, an* order-sorted $\Sigma$-algebra $A$ *is a many-sorted* $(S, \Sigma)$*-algebra* $A$ *such that:*

- *whenever* $s \leqslant s'$*, then we have* $A_s \subseteq A_{s'}$*, and*
- *whenever* $f : w \to s, f : w' \to s' \in f^{[s_1]\ldots[s_n]}_{[s]}$ *and* $\overline{a} \in A^w \cap A^{w'}$*, then we have* $A_{f:w\to s}(\overline{a}) = A_{f:w'\to s'}(\overline{a})$*, where* $A^{\epsilon} = 1$ *(*$\epsilon$ *denotes the empty string and* $1 = \{0\}$ *is a singleton set), and* $A^{s_1 \cdots s_n} = A_{s_1} \times \ldots \times A_{s_n}$*.*

*An* order-sorted $\Sigma$-homomorphism $h : A \to B$ *is a many-sorted* $(S, \Sigma)$*-homomorphism such that whenever* $[s] = [s']$ *and* $a \in A_s \cap A_{s'}$*, then we have* $h_s(a) = h_{s'}(a)$*. Call* $h$ injective*, resp.* surjective*, resp.* bijective*, iff for each* $s \in S$ $h_s$ *is injective, resp. surjective, resp. bijective. Call* $h$ *an* isomorphism *if there is another order-sorted* $\Sigma$*-homomorphism* $g : B \to A$ *such that for each* $s \in S$*,* $h_s; g_s = 1_{A_s}$*, and* $g_s; h_s = 1_{B_s}$*, with* $1_{A_s}, 1_{B_s}$ *the identity functions on* $A_s, B_s$*. This defines a category* $\mathbf{OSAlg}_{\Sigma}$*.* □

**Theorem 1.** *[73] The category* $\mathbf{OSAlg}_\Sigma$ *has an initial algebra. Furthermore, if* $\Sigma$ *is sensible, then the term algebra* $T_\Sigma$ *with:*

- *if* $a : \epsilon \to s$ *then* $a \in T_{\Sigma,s}$,
- *if* $t \in T_{\Sigma,s}$ *and* $s \leqslant s'$ *then* $t \in T_{\Sigma,s'}$,
- *if* $f : s_1 \dots s_n \to s$ *and* $t_i \in T_{\Sigma,s_i}$ $1 \leqslant i \leqslant n$, *then* $f(t_1, \dots, t_n) \in T_{\Sigma,s}$,

*is initial, i.e., there is a unique* $\Sigma$*-homomorphism from* $T_\Sigma$ *to each* $\Sigma$*-algebra.*

$T_\Sigma$ will (ambiguously) denote both the above-defined $S$-sorted set and the set $T_\Sigma = \bigcup_{s \in S} T_{\Sigma,s}$. For $[s] \in \widehat{S}$, $T_{\Sigma,[s]} = \bigcup_{s' \in [s]} T_{\Sigma,s'}$. An OS signature $\Sigma$ is said to *have non-empty sorts* iff for each $s \in S$, $T_{\Sigma,s} \neq \emptyset$. Unless explicitly stated otherwise, *I will assume throughout that* $\Sigma$ *has non-empty sorts.* An OS signature $\Sigma$ is called *preregular* [54] iff for each $t \in T_\Sigma$ the set $\{s \in S \mid t \in T_{\Sigma,s}\}$ has a least element, denoted $ls(t)$. *I will assume throughout that* $\Sigma$ *is preregular.*

An $S$-sorted set $X = \{X_s\}_{s \in S}$ of *variables*, satisfies $s \neq s' \Rightarrow X_s \cap X_{s'} = \emptyset$, and the variables in $X$ are always assumed disjoint from all constants in $\Sigma$. The $\Sigma$*-term algebra* on variables $X$, $T_\Sigma(X)$, is the *initial algebra* for the signature $\Sigma(X)$ obtained by adding to $\Sigma$ the variables $X$ *as extra constants.* Since a $\Sigma(X)$-algebra is just a pair $(A, \alpha)$, with $A$ a $\Sigma$-algebra, and $\alpha$ an *interpretation of the constants* in $X$, i.e., an $S$-sorted function $\alpha \in [X \to A]$, the $\Sigma(X)$-initiality of $T_\Sigma(X)$ can be expressed as the following corollary of Theorem 1:

**Theorem 2.** *(Freeness Theorem). If* $\Sigma$ *is sensible, for each* $A \in \mathbf{OSAlg}_\Sigma$ *and* $\alpha \in [X \to A]$*, there exists a unique* $\Sigma$*-homomorphism,* $\_\alpha : T_\Sigma(X) \to A$ *extending* $\alpha$*, i.e., such that for each* $s \in S$ *and* $x \in X_s$ *we have* $x\alpha_s = \alpha_s(x)$.

In particular, when $A = T_\Sigma(X)$, an interpretation of the constants in $X$, i.e., an $S$-sorted function $\sigma \in [X \to T_\Sigma(X)]$ is called a *substitution*, and its unique homomorphic extension $\_\sigma : T_\Sigma(X) \to T_\Sigma(X)$ is also called a substitution. Define $dom(\sigma) = \{x \in X \mid x \neq x\sigma\}$, and $ran(\sigma) = \bigcup_{x \in dom(\sigma)} vars(x\sigma)$. A *variable specialization* is a substitution $\rho$ that just renames a few variables and may lower their sort. More precisely, $dom(\rho)$ is a finite set of variables $\{x_1, \dots, x_n\}$, with respective sorts $s_1, \dots, s_n$, and $\rho$ injectively maps the $x_1, \dots, x_n$ to variables $x'_1, \dots, x'_n$ with respective sorts $s'_1, \dots, s'_n$ such that $s'_i \leqslant s_i$, $1 \leqslant i \leqslant n$.

The first-order language of *equational* $\Sigma$*-formulas* is defined in the usual way: its atoms are $\Sigma$*-equations* $t = t'$, where $t, t' \in T_\Sigma(X)_{[s]}$ for some $[s] \in \widehat{S}$ and each $X_s$ is assumed countably infinite. The set $Form(\Sigma)$ of *equational* $\Sigma$*-formulas* is then inductively built from atoms by: conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and universal ($\forall x{:}s$) and existential ($\exists x{:}s$) quantification with sorted variables $x{:}s \in X_s$ for some $s \in S$. The literal $\neg(t = t')$ is denoted $t \neq t'$.

Given a $\Sigma$-algebra $A$, a formula $\varphi \in Form(\Sigma)$, and an assignment $\alpha \in [Y \to A]$, with $Y = fvars(\varphi)$ the free variables of $\varphi$, the *satisfaction relation* $A, \alpha \models \varphi$ is defined inductively as usual: for atoms, $A, \alpha \models t = t'$ iff $t\alpha = t'\alpha$; for Boolean connectives it is the corresponding Boolean combination of the satisfaction relations for subformulas; and for quantifiers: $A, \alpha \models (\forall x{:}s)\ \varphi$ (resp. $A, \alpha \models (\exists x{:}s)\ \varphi$) holds iff for all $a \in A_s$ (resp. some $a \in A_s$) we have $A, \alpha \uplus \{(x{:}s, a)\} \models \varphi$, where the assignment $\alpha \uplus \{(x{:}s, a)\}$ extends $\alpha$ by mapping $x{:}s$ to $a$. Finally, $A \models \varphi$

holds iff $A, \alpha \models \varphi$ holds for each $\alpha \in [Y \to A]$, where $Y = \mathit{fvars}(\varphi)$. We say that $\varphi$ is *valid* (or *true*) in $A$ iff $A \models \varphi$. We say that $\varphi$ is *satisfiable* in $A$ iff $\exists \alpha \in [Y \to A]$ such that $A, \alpha \models \varphi$, where $Y = \mathit{fvars}(\varphi)$. For a subsignature $\Omega \subseteq \Sigma$ and $A \in \mathbf{OSAlg}_\Sigma$, the *reduct* $A|_\Omega \in \mathbf{OSAlg}_\Omega$ agrees with $A$ in the interpretation of all sorts and operations in $\Omega$ and discards everything in $\Sigma - \Omega$. If $\varphi \in \mathit{Form}(\Omega)$ we have the equivalence $A \models \varphi \;\Leftrightarrow\; A|_\Omega \models \varphi$.

An OS *equational theory* is a pair $T = (\Sigma, E)$, with $E$ a set of $\Sigma$-equations. $\mathbf{OSAlg}_{(\Sigma,E)}$ denotes the full subcategory of $\mathbf{OSAlg}_\Sigma$ with objects those $A \in \mathbf{OSAlg}_\Sigma$ such that $A \models E$, called the $(\Sigma, E)$-*algebras*. $\mathbf{OSAlg}_{(\Sigma,E)}$ has an *initial algebra* $T_{\Sigma/E}$ [73]. Given $T = (\Sigma, E)$ and $\varphi \in \mathit{Form}(\Sigma)$, we call $\varphi$ $T$-*valid*, written $E \models \varphi$, iff $A \models \varphi$ for each $A \in \mathbf{OSAlg}_{(\Sigma,E)}$. We call $\varphi$ $T$-*satisfiable* iff there exists $A \in \mathbf{OSAlg}_{(\Sigma,E)}$ with $\varphi$ satisfiable in A. Note that $\varphi$ is $T$-*valid* iff $\neg\varphi$ is $T$-*unsatisfiable*.

The inference system in [73] is *sound and complete* for OS equational deduction, i.e., for any OS equational theory $(\Sigma, E)$, and $\Sigma$-equation $u = v$ we have an equivalence $E \vdash u = v \quad \Leftrightarrow \quad E \models u = v$. Deducibility $E \vdash u = v$ is often abbreviated as $u =_E v$ and called $E$-*equality*. A preregular signature $\Sigma$ is called $E$-*preregular* iff for each $u = v \in E$ and variable specialization $\rho$, $ls(u\rho) = ls(v\rho)$.

In the above logical notions there is only an *apparent* lack of predicate symbols: full order-sorted first-order logic can be *reduced* to order-sorted algebra and the above language of equational formulas. The essential idea is to view a predicate $p(x_1{:}s_1, \ldots, x_n{:}s_n)$ as a function symbol $p : s_1 \ldots s_n \to \mathit{Pred}$, with $\mathit{Pred}$, a new sort having a constant $tt$. An atomic formula $p(t_1, \ldots, t_n)$ is then expressed as the equation $p(t_1, \ldots, t_n) = tt$. Let me just give a few technical details. An *order-sorted first-order logic signature*, or just an OS-FO signature, is a pair $(\Sigma, \Pi)$ with $\Sigma$ an OS signature with set of sorts $S$, and $\Pi$ an $S^*$-indexed set $\Pi = \{\Pi_w\}_{w \in S^*}$ of *predicate symbols*. An OS $(\Sigma, \Pi)$-*model* $M$ is an OS $\Sigma$-algebra $M$ together with an $S^*$-indexed mapping $M_\_ : \Pi \to \{\mathcal{P}(M^w)\}_{w \in S^*}$ interpreting each $p \in \Pi_w$ as a subset $M_p \subseteq M^w$. Since $p$ can be overloaded, we sometimes write $M_{p_w} \subseteq M^w$. $M$ must also satisfy the additional condition that *overloaded predicates agree on common data*. That is, if $w \equiv_{\leqslant} w'$, $p \in \Pi_w$ and $p \in \Pi_{w'}$, then for any $\overline{a} \in M^w \cap M^{w'}$ we have $\overline{a} \in M_{p_w} \;\Leftrightarrow\; \overline{a} \in M_{p_{w'}}$. The language of *first-order* $(\Sigma, \Pi)$-*formulas* extends that of equational $\Sigma$-formulas by adding as atomic formulas predicate expressions of the form $p(t_1, \ldots, t_n)$, with $p \in \Pi_w$ and $(t_1, \ldots, t_n) \in T_\Sigma(X)^w$. The satisfaction relation is likewise extended by defining $M, \alpha \models p(t_1, \ldots, t_n)$ iff $(t_1\alpha, \ldots, t_n\alpha) \in M_p$.

The reduction to OS algebra is achieved as follows. We associate to an OS-FO signature $(\Sigma, \Pi)$ an OS signature $(\Sigma \cup \Pi)$ by the above-mentioned method of adding to $\Sigma$ a new sort $\mathit{Pred}$ with a constant $tt$ in its own separate connected component $\{\mathit{Pred}\}$, and viewing each $p \in \Pi_w$ as a function symbol $p : s_1 \ldots s_n \to \mathit{Pred}$. The reduction at the model level is now very simple: each OS $(\Sigma \cup \Pi)$-algebra $A$ defines a $(\Sigma, \Pi)$-model $A^\circ$ with $\Sigma$-algebra structure $A|_\Sigma$ and having for each $p \in \Pi_w$ the predicate interpretation $A^\circ_p = A^{-1}_{p:w \to \mathit{Pred}}(tt)$. The reduction at the formula level is also quite simple: we map a $(\Sigma, \Pi)$-formula $\varphi$ to an equational formula $\widetilde{\varphi}$, called its *equational version*, by just replacing each atom

$p(t_1, \ldots, t_n)$ by the equational atom $p(t_1, \ldots, t_n) = tt$. The *correctness* of this reduction is just the easy to check equivalence:

$$A^\circ \models \varphi \;\Leftrightarrow\; A \models \widetilde{\varphi}.$$

An OS-FO *theory* is just a pair $((\Sigma, \Pi), \Gamma)$, with $(\Sigma, \Pi)$ an OS-FO signature and $\Gamma$ a set of $(\Sigma, \Pi)$-formulas. Call $((\Sigma, \Pi), \Gamma)$ *equational* iff $(\Sigma \cup \Pi, \widetilde{\Gamma})$ is an OS equational theory. By the above equivalence and the completeness of OS equational logic such theories allow a sound and complete use of equational deduction also with predicate atoms. Note that if $((\Sigma, \Pi), \Gamma)$ is equational, it is a very simple type of theory in OS Horn Logic with Equality and therefore has an initial model $T_{\Sigma,\Pi,\Gamma}$ [53]. A useful, easy to check fact is that we have an identity: $T^\circ_{\Sigma \cup \Pi/\widetilde{\Gamma}} = T_{\Sigma,\Pi,\Gamma}$. I will give natural examples of OS-FO equational theories later in the paper.

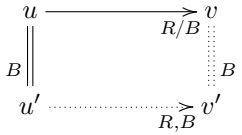
Recall the notation for term positions, subterms, and term replacement from [33]: (i) positions in a term viewed as a tree are marked by strings $p \in \mathbb{N}^*$ specifying a path from the root, (ii) $t|_p$ denotes the subterm of term $t$ at position $p$, and (iii) $t[u]_p$ denotes the result of *replacing* subterm $t|_p$ at position $p$ by $u$.

**Definition 3.** *A* rewrite theory *is a triple $\mathcal{R} = (\Sigma, B, R)$ with $(\Sigma, B)$ an order-sorted equational theory and $R$ a set of $\Sigma$-*rewrite rules*, i.e., sequents $l \to r$, with $l, r \in T_\Sigma(X)_{[s]}$ for some $[s] \in \widehat{S}$. In what follows it is always assumed that:*

1. *For each $l \to r \in R$, $l \notin X$ and $vars(r) \subseteq vars(l)$.*
2. *Each rule $l \to r \in R$ is* sort-decreasing*, i.e., for each variable specialization $\rho$, $ls(l\rho) \geqslant ls(r\rho)$.*
3. *$\Sigma$ is $B$-*preregular*.*
4. *Each equation $u = v \in B$ is* regular*, i.e., $vars(u) = vars(v)$, and* linear*, i.e., there are no repeated variables in $u$, and no repeated variables in $v$.*

*The* one-step *$R, B$-rewrite relation $t \to_{R,B} t'$, holds between $t, t' \in T_\Sigma(X)_{[s]}$, $[s] \in \widehat{S}$, iff there is a rewrite rule $l \to r \in R$, a substitution $\sigma \in [X \to T_\Sigma(X)]$, and a term position $p$ in $t$ such that $t|_p =_B l\sigma$, and $t' = t[r\sigma]_p$. Note that, by assumptions (2)–(3) above, $t[r\sigma]_p$ is always a well-formed $\Sigma$-term.*

*$\mathcal{R}$ is called: (i)* terminating *iff the relation $\to_{R,B}$ is well-founded; (ii)* strictly *$B$-*coherent *[71] iff whenever $u \to_{R,B} v$ and $u =_B u'$ there is a $v'$ such that $u' \to_{R,B} v'$ and $v =_B v'$ :*




*(iii)* confluent *iff $u \to^*_{R,B} v_1$ and $u \to^*_{R,B} v_2$ imply that there are $w_1, w_2$ such that $v_1 \to^*_{R,B} w_1$, $v_2 \to^*_{R,B} w_2$, and $w_1 =_B w_2$ (where $\to^*_{R,B}$ denotes the reflexive-transitive closure of $\to_{R,B}$); and (iv)* convergent *if (i)–(iii) hold. If $\mathcal{R}$ is convergent, for each $\Sigma$-term $t$ there is a term $u$ such that $t \to^*_{R,B} u$ and $(\nexists v)\ u \to_{R,B} v$.*

*We then write* $u = t!_{R,B}$ *and* $t \rightarrow !_{R,B} t!_{R,B}$*, and call* $t!_{R,B}$ *the* $R, B$-normal form *of* $t$*, which, by confluence, is unique up to* $B$*-equality.*

Given a set $E$ of $\Sigma$-equations, let $R(E) = \{u \rightarrow v \mid u = v \in E\}$. A *decomposition* of an order-sorted equational theory $(\Sigma, E)$ is a convergent rewrite theory $\mathcal{R} = (\Sigma, B, R)$ such that $E = E_0 \uplus B$ and $R = R(E_0)$. The key property of a decomposition is the following:

**Theorem 3.** *(Church-Rosser Theorem) [62,71] Let* $\mathcal{R} = (\Sigma, B, R)$ *be a decomposition of* $(\Sigma, E)$*. Then we have an equivalence:*

$$E \vdash u = v \;\; \Leftrightarrow \;\; u!_{R,B} =_B v!_{R,B}.$$

If $\mathcal{R} = (\Sigma, B, R)$ is a decomposition of $(\Sigma, E)$, and $X$ an $S$-sorted set of variables, the *canonical term algebra* $C_{\mathcal{R}}(X)$ has $C_{\mathcal{R}}(X)_s = \{[t!_{R,B}]_B \mid t \in T_\Sigma(X)_s\}$, and interprets each $f : s_1 \dots s_n \rightarrow s$ as the function $C_{\mathcal{R}}(X)_f : ([u_1]_B, \dots, [u_n]_B) \mapsto [f(u_1, \dots, u_n)!_{R,B}]_B$. By the Church-Rosser Theorem we then have an isomorphism $h : T_{\Sigma/E}(X) \cong C_{\mathcal{R}}(X)$, where $h : [t]_E \mapsto [t!_{R,B}]_B$. In particular, when $X$ is the empty family of variables, the canonical term algebra $C_{\mathcal{R}}$ is an initial algebra, and is the most intuitive possible model for $T_{\Sigma/E}$ as an algebra of *values* computed by $R, B$-simplification.

Quite often, the signature $\Sigma$ on which $T_{\Sigma/E}$ is defined has a natural decomposition as a disjoint union $\Sigma = \Omega \uplus \Delta$, where the elements of $C_{\mathcal{R}}$, that is, the *values* computed by $R, B$-simplification, are $\Omega$-terms, whereas the function symbols $f \in \Delta$ are viewed as *defined functions* which are *evaluated away* by $R, B$-simplification. $\Omega$ (with same poset of sorts as $\Sigma$) is then called a *constructor subsignature* of $\Sigma$. Call a decomposition $\mathcal{R} = (\Sigma, B, R)$ of $(\Sigma, E)$ *sufficiently complete* with respect to the *constructor subsignature* $\Omega$ iff for each $t \in T_\Sigma$ we have: (i) $t!_{R,B} \in T_\Omega$, and (ii) if $u \in T_\Omega$ and $u =_B v$, then $v \in T_\Omega$. This ensures that for each $[u]_B \in C_{\mathcal{R}}$ we have $[u]_B \subseteq T_\Omega$. Of course, we want $\Omega$ *as small as possible* with these properties. I give in what follows many examples of such decompositions $\Sigma = \Omega \uplus \Delta$ into constructors and defined functions. In Example 1 below, $\Omega = \{\top, \bot\}$ and $\Delta = \{\_ \wedge \_, \_ \vee \_\}$. Tools based on tree automata [28], equational tree automata [59], or narrowing [57], can be used to automatically check sufficient completeness of a decomposition $\mathcal{R}$ with respect to constructors $\Omega$ under some assumptions.

As the following definition shows, sufficient completeness is closely related to the notion of a *protecting* theory inclusion, which is itself a special case of an *extending* theory inclusion.

**Definition 4.** *An equational theory* $(\Sigma, E)$ protects *(resp.* extends*) another theory* $(\Omega, E_\Omega)$ *iff* $(\Omega, E_\Omega) \subseteq (\Sigma, E)$ *and the unique* $\Omega$*-homomorphism* $h : T_{\Omega/E_\Omega} \rightarrow T_{\Sigma/E}|_\Omega$ *is an isomorphism* $h : T_{\Omega/E_\Omega} \cong T_{\Sigma/E}|_\Omega$ *(resp. is injective).*

*A decomposition* $\mathcal{R} = (\Sigma, B, R)$ protects *(resp. is a* conservative extension of*) another decomposition* $\mathcal{R}_0 = (\Sigma_0, B_0, R_0)$ *iff* $\mathcal{R}_0 \subseteq \mathcal{R}$*, i.e.,* $\Sigma_0 \subseteq \Sigma$*,* $B_0 \subseteq B$*, and* $R_0 \subseteq R$*, and for all* $t, t' \in T_{\Sigma_0}(X)$ *we have: (i)* $t =_{B_0} t' \Leftrightarrow t =_B t'$*, (ii)* $t = t!_{R_0,B_0} \Leftrightarrow t = t!_{R,B}$*, and (iii)* $C_{\mathcal{R}_0} = C_{\mathcal{R}}|_{\Sigma_0}$ *(resp.* $C_{\mathcal{R}_0} \subseteq C_{\mathcal{R}}|_{\Sigma_0}$*).*

$\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ *is a* constructor decomposition *of* $\mathcal{R} = (\Sigma, B, R)$ *iff* $\mathcal{R}$ *protects* $\mathcal{R}_\Omega$ *and* $\Sigma$ *and* $\Omega$ *have the same poset of sorts, so that by (iii) above* $\mathcal{R}$ *is sufficiently complete with respect to* $\Omega$*. Furthermore,* $\Omega$ *is called a subsignature of* free constructors modulo $B_\Omega$ *iff* $R_\Omega = \emptyset$*, so that* $C_{\mathcal{R}_\Omega} = T_{\Omega/B_\Omega}$.

The case where all constructor terms are in $R, B$-normal form is captured by $\Omega$ being a subsignature of free constructors modulo $B_\Omega$. Note also that conditions (i) and (ii) are, so called, "no confusion" conditions, and for protecting extensions (iii) is a "no junk" condition, that is, $\mathcal{R}$ does not add new data to $C_{\mathcal{R}_0}$, whereas for conservative extensions (iii) is relaxed to the "no confusion" condition $C_{\mathcal{R}_0} \subseteq C_{\mathcal{R}}|_{\Sigma_0}$, which is already implicit in (i) and (ii). Therefore, protecting extensions are a stronger kind of conservative extensions.

Given an OS equational theory $(\Sigma, E)$ and a system of $\Sigma$-equations, that is, a conjunction $\phi = u_1 = v_1 \wedge \ldots \wedge u_n = v_n$ of $\Sigma$-equations, an *$E$-unifier* of it is a substitution $\sigma$ such that $u_i\sigma =_E v_i\sigma$, $1 \leqslant i \leqslant n$. An *$E$-unification algorithm* for $(\Sigma, E)$ is an algorithm generating a *complete set* of $E$-unifiers $Unif_E(\phi)$ for any system of $\Sigma$ equations $\phi$, where "complete" means that for any $E$-unifier $\sigma$ of $\phi$ there is a $\tau \in Unif_E(\phi)$ and a substitution $\rho$ such that $\sigma =_E \tau\rho$, where $=_E$ here means that for any variable $x$ we have $x\sigma =_E x\tau\rho$. Such an algorithm is called *finitary* if it always terminates with a *finite set* $Unif_E(\phi)$ for any such $\phi$.

The notion of *variant* answers, in a sense, two questions: (i) how can we best describe symbolically the elements of $C_{\mathcal{R}}(X)$ that are *reduced substitution instances* of a *pattern term* $t$? and (ii) given an original pattern $t$, how many other patterns do we need to describe the reduced instances of $t$ in $C_{\mathcal{R}}(X)$?

**Definition 5.** *Given a decomposition* $\mathcal{R} = (\Sigma, B, R)$ *of an OS equational theory* $(\Sigma, E)$ *and a* $\Sigma$*-term* $t$*, a* variant[2] *[32,43] of* $t$ *is a pair* $(u, \theta)$ *such that: (i)* $u =_B (t\theta)!_{R,B}$*, (ii) if* $x \notin vars(t)$*, then* $x\theta = x$*, and (iii)* $\theta = \theta!_{R,B}$*, that is,* $x\theta = (x\theta)!_{R,B}$ *for all variables* $x$*.* $(u, \theta)$ *is called a* ground variant *iff, furthermore,* $u \in T_\Sigma$*. Note that if* $(u, \theta)$ *is a ground variant of some* $t$*, then* $[u]_B \in C_{\mathcal{R}}$*. Given variants* $(u, \theta)$ *and* $(v, \gamma)$ *of* $t$*,* $(u, \theta)$ *is called* more general *than* $(v, \gamma)$*, denoted* $(u, \theta) \sqsupseteq_{R,B} (v, \gamma)$*, iff there is a substitution* $\rho$ *such that: (i)* $\theta\rho =_B \gamma$*, and (ii)* $u\rho =_B v$*. Let* $[\![t]\!]_{R,B} = \{(u_i, \theta_i) \mid i \in I\}$ *denote a* most general complete set of variants *of* $t$*, that is, a set of variants such that: (i) for any variant* $(v, \gamma)$ *of* $t$ *there is an* $i \in I$*, such that* $(u_i, \theta_i) \sqsupseteq_{R,B} (v, \gamma)$*; and (ii) for* $i, j \in I$*,* $i \neq j \Rightarrow ((u_i, \theta_i) \not\sqsupseteq_{R,B} (u_j, \theta_j) \wedge (u_j, \theta_j) \not\sqsupseteq_{R,B} (u_i, \theta_i))$*. A decomposition* $\mathcal{R} = (\Sigma, B, R)$ *of* $(\Sigma, E)$ *has the* finite variant property *[32] (FVP) iff for each* $\Sigma$*-term* $t$ *there is a* finite *most general complete set of variants* $[\![t]\!]_{R,B} = \{(u_1, \theta_1), \ldots, (u_n, \theta_n)\}$.

If $B$ has a finitary unification algorithm, the *folding variant narrowing* strategy described in [43] provides an effective method to generate $[\![t]\!]_{R,B}$. Furthermore, $[\![t]\!]_{R,B}$ is finite for each $t$, so that the strategy *terminates*, iff $\mathcal{R}$ is FVP.

*Example 1.* (Booleans). Let $\mathcal{B} = (\Sigma, \emptyset, R)$ with $\Sigma$ having a single sort, say *Truth*, constants $\top, \bot$, a unary operation $\neg$, and binary opertors $\_ \wedge \_$ and $\_ \vee \_$,

[2] For a discussion of similar but not exactly equivalent versions of the variant notion see [24]. Here I follow the formulation in [43].

and $R$ the rules: $\neg(\top) \to \bot$, $\neg(\bot) \to \top$, $\top \wedge x \to x$, $\bot \wedge x \to \bot$, $\bot \vee x \to x$, and $\top \wedge x \to \top$. Then $\mathcal{B}$ is FVP. For example, $[\![x \wedge y]\!]_{R,B} = \{(x \wedge y, id), (y, \{x \mapsto \top\}), (\bot, \{x \mapsto \bot\})\}$.

FVP is a *semi-decidable* property [24], which can be easily verified (when it holds) by checking, using folding variant narrowing, that for each function symbol $f : s_1 \dots s_n \to s$ the term $f(x_1, \dots, x_n)$, with $x_i$ of sort $s_i$, $1 \leqslant i \leqslant n$, has a finite number of most general variants. Given an FVP decomposition $\mathcal{R}$ its *variant complexity* is the total number $n$ of variants for all such $f(x_1, \dots, x_n)$, provided $f$ has some associated rules of the form $f(t_1, \dots, t_n) \to t'$. This gives a *rough* measure of how costly it is to perform variant computations *relative* to the cost of performing $B$-unification. The variant complexity of $\mathcal{B}$ above is 9. The measure is rough: a more accurate one would be variant complexity per symbol. Five symbols, each with variant complexity 2, is much better than two symbols each with variant complexity 5. For $\mathcal{B}$, the variant complexity per symbol is 3.

Folding variant narrowing provides a method for generating a *complete set of $E$-unifiers*. I give below a method for generating such a set that is different from the one given in [43], because in Section 4 this will make it possible to express the notion of constructor $E$-unifier in a straightforward way. Let $(\Sigma, E)$ have a decomposition $\mathcal{R} = (\Sigma, B, R)$ with $B$ having a finitary $B$-unification algorithm.

To be able to express systems of equations, say, $u_1 = v_1 \wedge \dots \wedge u_n = v_n$, as *terms*, we can extend $\Sigma$ to a signature $\Sigma^\wedge$ by adding:

1. for each connected component $[s]$ that does not already have a top element, a fresh new sort $\top_{[s]}$ with $\top_{[s]} > s'$ for each $s \in [s]$. In this way we obtain a (possibly extended) poset of sorts $(S_\top, \geqslant)$;
2. fresh new sorts *Lit* and *Conj* with a subsort inclusion $Lit < Conj$, with a binary conjunction operator $\_ \wedge \_ : Lit\ Conj \to Conj$, and
3. for each connected component $[s] \in \widehat{S_\top}$ with top sort $\top_{[s]}$, binary operators $\_ = \_ : \top_{[s]}\ \top_{[s]} \to Lit$ and $\_ \neq \_ : \top_{[s]}\ \top_{[s]} \to Lit$.

**Theorem 4.** *Under the above assumptions on $\mathcal{R}$, let $\phi = u_1 = v_1 \wedge \dots \wedge u_n = v_n$ be a system of $\Sigma$-equations viewed as a $\Sigma^\wedge$-term of sort Conj. Then*

$$\{\theta\gamma \mid (\phi', \theta) \in [\![\phi]\!]_{R,B} \wedge \gamma \in Unif_B(\phi') \wedge (\phi'\gamma)!_{R,B} = \phi'\gamma \wedge (\theta\gamma)!_{R,B} = \theta\gamma\}$$

*is a complete set of $E$-unifiers for $\phi$, where $Unif_B(\phi')$ denotes a complete set of most general $B$-unifiers for each variant $\phi' = u'_1 = v'_1 \wedge \dots \wedge u'_n = v'_n$.*

*Proof.* First of all note that all the substitutions in the above set are $E$-unifiers by construction. Second, observe that if $\alpha$ is an $E$-unifier of $\phi$, then the $R, B$-normalized substitution $\alpha!_{R,B}$ is a unifier $E$-equivalent to $\alpha$. Therefore, we can assume without loss of generality that all unifiers $\alpha$ are $R, B$-normalized. We just need to show that any $R, B$-normalized unifier $\alpha$ is $B$-equivalent to an instance of one in the above set. But by the Church-Rosser Theorem such an $\alpha$ is an $E$-unifier of $\phi$ iff $(u_i\alpha)!_{R,B} =_B (v_i\alpha)!_{R,B}$, $1 \leqslant i \leqslant n$, iff: (i) $((\phi\alpha)!_{R,B}, \alpha)$ is an $R, B$-variant of $\phi$, and (ii) $(u_i\alpha)!_{R,B} =_B (v_i\alpha)!_{R,B}$, $1 \leqslant i \leqslant n$. But then there must be a $(\phi', \theta) \in [\![\phi]\!]_{R,B}$ such that $(\phi', \theta) \sqsupseteq_{R,B} ((\phi\alpha)!_{R,B}, \alpha)$. That is,

there is a $\beta$ such that: (i) $(\phi'\beta) =_B (\phi\alpha)!_{R,B}$, and (ii) $\theta\beta =_B \alpha$. But since $\beta$ $B$-unifies $\phi'$, there must be a $\gamma \in \mathit{Unif}_B(\phi')$ and a $\rho$ such that $\beta =_B \gamma\rho$, so that $\alpha =_B \theta\gamma\rho$. But: (i) $\alpha = \alpha!_{R,B}$ forces $\theta\gamma = (\theta\gamma)!_{R,B}$; and (ii) $(\phi'\beta) =_B (\phi\alpha)!_{R,B}$ and $\beta =_B \gamma\rho$ force $(\phi'\gamma)!_{R,B} = \phi'\gamma$. Therefore, the above set is a complete set of $E$-unifiers for $u = v$. □

Form now on, $\mathit{Unif}_E(\phi)$ will abbreviate the complete set of $E$-unifiers specified in Theorem 4, which will be called a complete set of *variant-based $E$-unifiers* of $\phi$. Since if $\mathcal{R} = (\Sigma, B, R)$ is FVP, then $\mathcal{R}^\wedge = (\Sigma^\wedge, B, R)$ is also FVP, Theorem 4 shows that if a finitary $B$-unification algorithm exists and $\mathcal{R}$ is an FVP decomposition of $(\Sigma, E)$, then $E$ has a finitary $E$-unification algorithm.

## 3 A Satisfiability Puzzle

In Section 8 of their paper about the finite variant property [32], Comon-Lundh and Delaune give a theorem (Theorem 3) stating that if $(\Sigma, E)$ has an FVP decomposition, say $\mathcal{R} = (\Sigma, E', R)$, and satisfiability of quantifier-free (QF) equational $\Sigma$-formulas in the initial algebra $T_{\Sigma/E'}$ is decidable,[3] then satisfiability of QF equational $\Sigma$-formulas in the initial algebra $T_{\Sigma/E}$ is also decidable. They give the following proof sketch for this theorem:

> *To prove this, simply compute the variants $\phi_1, \ldots, \phi_n$ of the formula $\phi$. (In such a computation, logical connectives are seen as free symbols). For every substitution $\sigma$, there is an index $i$ and a substitution $\theta$ such that $\phi\sigma!_{R,E'} =_{E'} \phi_i\theta$. In particular, $\phi$ is solvable modulo $E$ iff one of the $\phi_i$ is solvable modulo $E'$.*

The actual text in [32] only differs from the one above by the use of a different notation for the normal form $\phi\sigma!_{R,E'}$. Their theorem, however, is incorrect, as shown below. Since by putting a QF formula in DNF we can reduce satisfiability of a QF formula to satisfiability of a conjunction of literals, we can further simplify the above proof sketch by focusing on such conjunctions.

What the proof sketch then means is that, since $(\Sigma^\wedge, E)$ has an FVP decomposition $\mathcal{R}^\wedge = (\Sigma^\wedge, E', R)$, and each conjunction of literals, say, $\phi = B_1 \wedge \ldots \wedge B_k$, with each $B_i$ either a $\Sigma$-equation or a $\Sigma$-disequation, is a $\Sigma^\wedge$-term, the proof sketch is a claim that $\phi$ is satisfiable in $T_{\Sigma/E}$ iff for some $R, E'$-variant $(\phi_i, \theta_i)$ of $\phi$ the conjunction $\phi_i$ is satisfiable in $T_{\Sigma/E'}$.

*Example 2.* The following counterexample shows that Theorem 3 in [32] is incorrect. Let $\Sigma$ have sorts *Nat* and *Bool*, with constants 0 of sort *Nat* and $\top, \bot$ of sort *Bool*, a unary successor operator $s$ of sort *Nat*, and a unary $\mathit{zero?} : \mathit{Nat} \to \mathit{Bool}$. Let $n$ be a variable of sort *Nat*, and $E$ the equations $\mathit{zero?}(s(n)) = \bot$ and $\mathit{zero?}(0) = \top$. Then $\mathcal{N}_{\mathit{zero?}} = (\Sigma, \varnothing, R(E))$ is an FVP decomposition of $(\Sigma, E)$

[3] Such decidable QF satisfiability is of course equivalent to the decidability of whether a sentence in the existential closure of such QF formulas belongs to the theory of $T_{\Sigma/E'}$, which is how the decidability property is actually stated in [32].

of variant complexity 3 (i.e., in the above notation $E' = \varnothing$). Let $\phi$ be the formula $x = \mathit{zero}?(n) \ \wedge \ x \neq \top \ \wedge \ x \neq \bot$. It has a complete set of three most general $R(E), \varnothing$-variants, namely: $(\phi, \mathit{id})$, $(\phi', \{n \mapsto s(n')\})$, and $(\phi'', \{n \mapsto 0\})$, with $n'$ of sort $\mathit{Nat}$, $\mathit{id}$ the identity substitution, the other substitutions specified by how they map the variable $n$ in $\phi$, and where $\phi'$ is the formula $x = \bot \ \wedge \ x \neq \top \ \wedge \ x \neq \bot$, and $\phi''$ is the formula $x = \top \ \wedge \ x \neq \top \ \wedge \ x \neq \bot$. The formula $\phi$ is clearly unsatisfiable in $T_{\Sigma/E}$. However, for the variant $(\phi, \mathit{id})$ the formula $\phi$ *is* satisfiable in $T_\Sigma$ for *any* substitution $\sigma = \{n \mapsto t, x \mapsto \mathit{zero}?(t)\}$ with $t$ a ground term of sort $\mathit{Nat}$; for example for $\sigma = \{n \mapsto 0, x \mapsto \mathit{zero}?(0)\}$.

A question still remains: whether, under suitable conditions, some analogue of the (incorrect) Theorem 3 in [32] could somehow be obtained. That is, can we find some results relating satisfiability in the *initial* algebras $T_{\Sigma/E}$ and in $T_{\Sigma/B}$ (or some initial algebra related to $T_{\Sigma/B}$) when $\mathcal{R} = (\Sigma, B, R)$ is an FVP decomposition of $(\Sigma, E)$? I address this question in Sections 5–7. The key to answer the question is the new notion of constructor variant that I present next.

## 4 Constructor Variants and Constructor Unifiers

Intuitively, an $R, B$-variant of a term $t$ is another term $v$ which is the normal form of an instance $t\theta$ of $t$; i.e., such variants $v$ are *patterns* covering the normal forms of instances of $t$. We can ask: what variants of $t$ cover (i.e., have as instances) the ground variants? I call such variants the *constructor variants* of $t$.

Likewise, given a system of equations $\phi$, call a $R, B$-normalized unifier $\theta$ of $\phi$ a *ground unifier* of $\phi$ iff $\theta$ is a ground substitution. And we can ask: what unifiers of $\phi$ cover (i.e., have as instances) the ground unifiers? I call such unifiers the *constructor unifiers* of $\phi$.

**Definition 6.** *Let $\mathcal{R} = (\Sigma, B, R)$ be a decomposition of $(\Sigma, E)$, and let $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ be a constructor decomposition of $\mathcal{R}$. Then an $R, B$-variant $(u, \theta)$ of a $\Sigma$-term $t$ is called a* constructor $R, B$-variant *of $t$ iff $u \in T_\Omega(X)$.*

*Suppose, furthermore, that $B$ has a finitary $B$-unification algorithm, so that, given a unification problem $\phi = u_1 = v_1 \wedge \ldots \wedge u_n = v_n$, Theorem 4 allows us to generate a complete set $\mathit{Unif}_E(\phi)$ of $E$-unifiers of $\phi$ by folding variant narrowing with $\mathcal{R}^\wedge$. We call a $R, B$-normalized substitution $\theta = \theta!_{R,B}$ with $\mathit{dom}(\phi) = \mathit{vars}(\phi)$ a* constructor $E$-unifier *of $\phi$ iff $(u_i\theta)!_{R,B} \in T_\Omega(X)$, $1 \leqslant i \leqslant n$.*

That these are the *right* definitions, in the sense that constructor variants (resp. constructor unifiers) do *cover* the ground variants (resp. ground unifiers) is shown in the following theorem.

**Theorem 5.** *(Completeness of Constructor Variants and Constructor Unifiers). Let $\mathcal{R} = (\Sigma, B, R)$ be a decomposition of $(\Sigma, E)$, and $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ a constructor decomposition of $\mathcal{R}$, and let $\phi = u_1 = v_1 \wedge \ldots \wedge u_n = v_n$ be a system of $\Sigma$-equations with $Y = \mathit{vars}(\phi)$. Then:*

1. *(Completeness of Constructor Variants). For each ground variant* $(u, \alpha)$ *of a* $\Sigma$*-term* $t$*, there exists a constructor variant* $(v, \beta)$ *and a substitution* $\rho$ *such that* $u =_B v\rho$ *and* $\alpha =_B \beta\rho$.
2. *(Completeness for Constructor Unifiers). If* $\delta \in [Y \to T_\Omega]$ *is a* $R, B$*-normalized ground* $E$*-unifier of* $\phi$*, then there is a constructor* $E$*-unifier* $\theta$ *of* $\phi$ *and a substitution* $\gamma$ *such that* $\delta =_B \theta\gamma$.
3. *(Unifiability).* $T_{\Sigma/E}(X) \models (\exists Y)\ \phi$ *iff* $T_{\Sigma/E} \models (\exists Y)\ \phi$ *iff* $\phi$ *has a constructor* $E$*-unifier. Furthermore, we have equivalences:*

$$E \models (\exists Y)\ \phi \ \Leftrightarrow\ T_{\Sigma/E}(X) \models (\exists Y)\ \phi \ \Leftrightarrow\ T_{\Sigma/E} \models (\exists Y)\ \phi.$$

*Proof.* (1) is trivial, since any ground variant is a constructor variant, so we can take $(u, \alpha) = (v, \beta)$, and $\rho$ the identity substitution. (2) is also trivial, since by sufficient completeness of the constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ any $R, B$-normalized ground unifier of $\phi$ is a constructor unifier of $\phi$, so we can take $\theta = \delta$ and $\gamma$ the identity substitution. To see (3), note that $T_{\Sigma/E} \models (\exists Y)\ \phi$ holds iff $\phi$ has a ground $E$-unifier $\delta$, which in particular is an $E$-unifier, so that $T_{\Sigma/E}(X) \models (\exists Y)\ \phi$ also holds. Conversely, if $T_{\Sigma/E}(X) \models (\exists Y)\ \phi$ holds, as witnessed by a unifier $\theta$, since $\Sigma$ has non-empty sorts, $[Y \to T_\Omega]$ is non-empty, therefore, $(\theta\gamma)!_{R,B}$ is a ground $E$-unifier of $\phi$ (and a fortiori a constructor $E$-unifier) for any $\gamma \in [Y \to T_\Omega]$, so that $T_{\Sigma/E} \models (\exists Y)\ \phi$ holds. This gives us the equivalence $T_{\Sigma/E}(X) \models (\exists Y)\ \phi \ \Leftrightarrow\ T_{\Sigma/E} \models (\exists Y)\ \phi$. And we have the implication $E \models (\exists Y)\ \phi \ \Rightarrow\ T_{\Sigma/E} \models (\exists Y)\ \phi$. But if $T_{\Sigma/E} \models (\exists Y)\ \phi$ holds, say thanks to a ground $E$-unifier $\delta$, and $A \models E$, then $A \models (\exists Y)\ \phi$ with witness $\delta h$, where $h : T_\Sigma \to A$ is the unique $\Sigma$-homomorphism from the initial algebra $T_\Sigma$. Therefore, we have $T_{\Sigma/E} \models (\exists Y)\ \phi \ \Rightarrow\ E \models (\exists Y)\ \phi$, as desired. □

It follows from Theorem 5 that if we are only interested in *E-unifiability* of a system of equations $\phi$ (that is, in the *satisfiability* of $\phi$ in $T_{\Sigma/E}$), only constructor $E$-unifiers are needed. Likewise, if we are only interested in patterns (variants) that describe ground terms in canonical form, i.e., terms up to $B_\Omega$-equivalence in the canonical term algebra $C_\mathcal{R}$, then only constructor variants are needed.

But are *all* constructor variants needed? That is, in the above theorem, we can ask the opposite question about (2): given a constructor variant $(v, \beta)$, is there a ground variant $(u, \alpha)$ and a substitution $\rho$ such that $u =_B v\rho$ and $\alpha =_B \beta\rho$ (so that $(v, \beta)$ is indeed "needed")? The answer is that in some decompositions *some constructor variants may not have any ground variant instances* and may not be "needed" in the above sense, as shown in the following example.

*Example 3.* Let $\Sigma$ be the unsorted signature with constant 0 and unary symbol $s$, and $E$ the single equation $s(s(0)) = 0$. Then $\mathcal{N}_2 = (\Sigma, \varnothing, R(E))$ is a decomposition of $(\Sigma, E)$ because: (i) $R(E)$ is size-decreasing and therefore terminating; and (ii) $R(E)$ has no critical pairs and therefore is locally confluent. Furthermore, $(\Sigma, \varnothing, R(E))$ is FVP of variant complexity 2, because the term $s(x)$ has two most general variants, namely, $(s(x), id)$, and $(0, \{x \mapsto s(0)\})$. Its only constructor decomposition is $(\Sigma, \varnothing, R(E))$ itself. In particular, $(s(s(y)), \{x \mapsto s(y)\})$ is a constructor variant of $s(x)$. But there is no ground variant $(u, \alpha)$ of $s(x)$ and substitution $\rho$ such that $u = s(s(y))\rho$.

An important question is how to *compute complete* sets of constructor $R, B$-variants of a term $t$, and of constructor unifiers of a system of equations $\phi$; and under what conditions such sets can be finite. Let me define these notions. Call a set $[\![t]\!]^{\Omega}_{R,B}$ of constructor variants of a term $t$ *complete* iff for each constructor variant $(v, \beta)$ of $t$ there is a $(w, \alpha) \in [\![t]\!]^{\Omega}_{R,B}$ and a substitution $\gamma$ such that $v =_B w\gamma$, and $\beta =_B \alpha\gamma$. Likewise, call a set $Unif^{\Omega}_{E}(\phi)$ of constructor unifiers of a system of equations $\phi$ *complete* iff for each constructor unifier $\theta$ of $\phi$ there is a $\rho \in Unif^{\Omega}_{E}(\phi)$ and a substitution $\gamma$ such that $(\phi\theta)!_{R,B} =_B (\phi\rho)!_{R,B}\gamma$, and $\theta =_B \rho\gamma$.

The easiest answer to this question can be given when $\mathcal{R} = (\Sigma, B, R)$ is a decomposition of $(\Sigma, E)$, $\mathcal{R}_{\Omega} = (\Omega, B_{\Omega}, R_{\Omega})$ is a constructor decomposition of $\mathcal{R}$, $\Sigma$ is a *many-sorted* signature, and $B$ is a combination of associativity, commutativity and identity axioms such that no binary operator $f \in \Sigma - \Omega$ has an identity axiom in $B$. Then, the computation of a *complete set of constructor variants*, denoted $[\![t]\!]^{\Omega}_{R,B}$, is easy: $[\![t]\!]^{\Omega}_{R,B} = \{(u, \alpha) \mid (u, \alpha) \in [\![t]\!]_{R,B} \wedge u \in T_{\Omega}(X)\}$. This set is complete because if $(v, \beta)$ is a constructor variant of $t$ there is a $(u, \alpha) \in [\![t]\!]_{R,B}$ and a substitution $\gamma$ such that $v =_B u\gamma$, and $\beta =_B \alpha\gamma$. But since $v$ is a constructor term and $\Sigma$ is many-sorted, this forces $u$ to be a constructor term, because if $u$ were to contain a symbol $f \in \Sigma - \Omega$, since no identity axioms hold for such symbols, $f$ could never disappear from $u\gamma$ by application of any axioms in $B$. Therefore, $(u, \alpha) \in [\![t]\!]^{\Omega}_{R,B}$. Furthermore, if $\mathcal{R}$ is FVP, the set $[\![t]\!]^{\Omega}_{R,B}$ is finite.

Likewise, the computation of a *complete set of constructor unifiers* of $\phi = u_1 = v_1 \wedge \ldots \wedge u_n = v_n$, denoted $Unif^{\Omega}_{E}(\phi)$ is also easy, namely, $Unif^{\Omega}_{E}(\phi)$ is the set

$$\{\theta\gamma \mid (\phi', \theta) \in [\![\phi]\!]^{\Omega^{\wedge}}_{R,B} \wedge \gamma \in Unif_{B}(\phi') \wedge (\phi'\gamma)!_{R,B} = \phi'\gamma \wedge (\theta\gamma)!_{R,B} = \theta\gamma\}$$

which, since $[\![\phi]\!]^{\Omega^{\wedge}}_{R,B} \subseteq [\![\phi]\!]_{R,B}$, is a subset of $Unif^{\Omega}_{E}(\phi)$. This set is again complete, because if $\alpha$ is a constructor unifier of $\phi$, then, by the proof of Theorem 4 we have a $\theta\gamma \in Unif_{E}(\phi)$ and a substitution $\delta$ such that $(\phi\theta\gamma)!_{R,B}\delta =_B (\phi\alpha)!_{R,B}$, and $\theta\gamma\delta =_B \alpha$. But since $((\phi\alpha)!_{R,B}, \alpha)$ is a constructor variant of $\phi$, $(\phi\theta\gamma)!_{R,B}\delta =_B (\phi\alpha)!_{R,B}$ $\Sigma^{\wedge}$ has no subtype-polymorphic operators, and $B$ has no identity axioms for any $f \in \Sigma - \Omega$, $((\phi\theta\gamma)!_{R,B}, \theta\gamma)$ must be a constructor variant of $\phi$ as well, so that $\theta\gamma \in Unif^{\Omega}_{E}(\phi)$. Again, if $\mathcal{R}$ is FVP, then the set $Unif^{\Omega}_{E}(\phi)$ is finite. Let us see some examples.

*Example 4.* Let $(\Sigma, E)$ be the many-sorted equational theory of Example 2 and $\mathcal{R} = (\Sigma, \varnothing, R(E))$ its associated FVP decomposition. It has a *free* constructor decomposition $(\Omega, \varnothing, \varnothing)$ with $\Omega$ consisting of $0, \top, \bot$ and $s$. The term $zero?(n)$ has three variants: $(zero?(n), id)$, $(\bot, \{n \mapsto s(n')\})$, and $(\top, \{n \mapsto 0\})$. Since $zero?(n)$ is *not* an $\Omega$-term, only the last two are constructor variants, defining the set $[\![zero?(n)]\!]^{\Omega}_{R,B}$.

The $E$-unification problem $zero?(n) = zero?(m)$ has three most general unifiers: $\{n \mapsto m\}$, $\{n \mapsto s(n'), m \mapsto s(m')\}$, and $\{n \mapsto 0, m \mapsto 0\}$. Only the last two are constructor unifiers, defining the set $Unif^{\Omega}_{E}(zero?(n) = zero?(m))$.

*Example 5.* Consider the unsorted theory $(\Sigma, E)$ where $\Sigma$ has a constant 0, a unary $s$ and a binary $\_+\_$, and $E$ has the equations $n+0 = n$, $n+s(m) = s(n+m)$. $(\Sigma, E)$ is not FVP, but it has an obvious decomposition $\mathcal{R} = (\Sigma, \emptyset, R(E))$. It furthermore has a *free* constructor decomposition $(\Omega, \emptyset, \emptyset)$ with $\Omega = \{0, s\}$.

The variants of the term $x + y$ are of the following types: (i) $(x + y, id)$, (ii) $(x, \{y \mapsto 0\})$, (iii) $(s^n(x+y'), \{y \mapsto s^n(y')\})$, $n \geqslant 1$, and (iv) $(s^n(x), \{y \mapsto s^n(0)\})$, $n \geqslant 1$. Only variants of types (ii) and (iv) are constructor variants, defining the set $[\![x + y]\!]^{\Omega}_{R,B}$.

The $E$-unification problem $x+y = z+0$ has the following types of $E$-unifiers: (i) $\{z \mapsto x + y\}$, (ii) $\{z \mapsto x, y \mapsto 0\}$, (iii) $\{z \mapsto s^n(x + y'), y \mapsto s^n(y')\}$, and (iv) $\{z \mapsto s^n(x), y \mapsto s^n(0)\}$. Only unifiers of types (ii) and (iv) are constructor unifiers, defining the set $Unif^{\Omega}_{E}(x + y = z + 0)$.

As the above examples show, there can be considerably fewer constructor variants and constructor $E$-unifiers than general variants and general $E$-unifiers. This means that using constructor variants and unifiers can be considerably more efficient for various purposes. For example, it can yield a considerably smaller search space in various automated deduction problems. For many-sorted signatures there is the added advantage that filtering constructor variants (resp. constructor unifiers) from a set of variants (resp. unifiers) is easy to do and inexpensive.

The issue of how to compute complete sets $[\![t]\!]^{\Omega}_{R,B}$, resp. $Unif^{\Omega}_{E}(\phi)$, of constructor variants, resp. constructor unifiers, when $\Sigma$ is *order-sorted* is more subtle, because in such a case, due to subsort polymorphism, the same symbol $f$ can have some typings in a constructor subsignature $\Omega$ and other, non-constructor typings in $\Sigma - \Omega$. Let us see a simple example illustrating the issues involved. The example also shows how a specification, such as that of Example 3, that *cannot* have a signature of free constructors at the unsorted or many-sorted levels, may in fact have a signature of *free* constructors in a more expressive order-sorted version of the original signature.

*Example 6.* Consider the following order-sorted version of Example 3, where now $\Sigma$ has sorts *Zero*, *One* and *Nat*, subsort inclusions $Zero, One < Nat$, 0 has sort *Zero*, and $s$ is subsort-polymorphic with typings $s : Zero \to One$ and $s : Nat \to Nat$. Again, $E$ consists of the single equation $s(s(0)) = 0$, and $\mathcal{N}^{os}_2 = (\Sigma, \emptyset, R(E))$ is an FVP decomposition of $(\Sigma, E)$ with variant complexity 2. But now we have a subsignature $\Omega$ of *free* constructors, obtained by just removing from $\Sigma$ the operator declaration $s : Nat \to Nat$.

Consider, as before, the term $s(x)$, with $x$ of sort *Nat*. Notice three things:

1. The variant $(s(s(y)), \{x \mapsto s(y)\})$, which in Example 3 was a constructor variant with no ground variant instances now is *not* a constructor variant at all. Furthermore, it has *no instances* that are constructor variants.
2. The complete set $[\![s(x)]\!]^{\Omega}_{R,B}$ of constructor variants of $s(x)$ is $[\![s(x)]\!]^{\Omega}_{R,B} = \{(0, \{x \mapsto s(0)\}), (s(z), \{x \mapsto z\})\}$, where $z$ has sort *Zero*.
3. The substitution $\{x \mapsto z\}$ in the second constructor variant $(s(z), \{x \mapsto z\})$ is the most general syntactic unifier of the equation $s(x) = x'$ with $x'$ of sort *One*, and $s(z)$ is the substitution instance of $s(x)$ under that substitution.

Points (1) and (3) give some hints about how the computation of the sets $[\![t]\!]^{\Omega}_{R,B}$ and $Unif^{\Omega}_{E}(\phi)$ can be carried out in the order-sorted case: computing them essentially reduces to computing certain simple $B$-*unifiers* of the form $t = y$ with $y$ having a suitable subsort of the least sort $ls(t)$ of $t$. Such unifiers can *specialize* some non-constructor terms like $s(x)$ to constructor instances like $s(z)$. In this example $B = \varnothing$, and the crucial reason why the unification problem $s(x) = x'$ arises is the typing $s : Zero \to One$ in the constructor subsignature $\Omega$. Also, the crucial reason why $(s(s(y)), \{x \mapsto s(y)\})$ has no constructor variant instances is that the unification problem $(s(s(y)) = x'$ with $x'$ of sort $One$ has no unifiers. A detailed description of how the computations of $[\![t]\!]^{\Omega}_{R,B}$ and $Unif^{\Omega}_{E}(\phi)$ are reduced to such simple $B$-unification problems is outside the scope of this paper. It can be found, together with a description of a prototype Maude implementation, in [85,86]. The above hints give some key intuitions but simplify the picture somewhat: in general the relevant unification problems cannot be solved in either $\Sigma$ or $\Omega$ alone. They need a transformed signature combining both $\Sigma$ and $\Omega$.

For the purposes of the present paper, all we need to know about the effective construction of $[\![t]\!]^{\Omega}_{R,B}$ and $Unif^{\Omega}_{E}(\phi)$ in [85,86], where $(\Sigma, B, R)$ is a decomposition of $(\Sigma, E)$ and $B$ has a finitary unification algorithm, is that if $(\Sigma, B, R)$ is FVP, then $[\![t]\!]^{\Omega}_{R,B}$ and $Unif^{\Omega}_{E}(\phi)$ are *finite* sets.

## 5 Satisfiability in Initial Algebras: Descent Results

Using the constructor variant notion from Section 4 we can associate the failure of Theorem 3 of [32] in Example 2 to the fact that for $\phi$ the formula $x = zero?(n) \wedge x \neq \top \wedge x \neq \bot$, the variant $(\phi, id)$ is *not* a constructor variant. One might conjecture that if $\mathcal{R} = (\Sigma, B, R)$ is an FVP decomposition of $(\Sigma, E)$, a QF equational formula $\phi$ is satisfiable in $T_{\Sigma/E}$ iff for some *constructor variant* $(\phi', \theta)$ $\phi'$ is satisfiable in $T_{\Sigma/B}$. But this conjecture fails in general:

*Example 7.* Let $(\Sigma, E)$ be the unsorted theory in Example 3, where $E$ consist of the single equation $s(s(0)) = 0$ and $\mathcal{N}_2 = (\Sigma, \varnothing, R(E))$ is an FVP decomposition of $(\Sigma, E)$. Let $\phi$ be the formula $x \neq 0 \wedge x \neq s(0)$. Its only $R(E), \varnothing$-variant is $(\phi, id)$, which, since in this case $\Sigma = \Omega$, is a constructor variant. Furthermore, this constructor variant has ground variant instances such as, for example, the ground variant $(0 \neq 0 \wedge 0 \neq s(0), \{x \mapsto 0\})$. Obviously, $\phi$ is unsatisfiable in $T_{\Sigma/E}$, but it is clearly satisfiable in $T_{\Sigma/\varnothing} = T_{\Sigma}$, for example with the ground substitution $\{x \mapsto s(s(0))\}$. Of course, since $T_{\Sigma/E}$ is a *finite* algebra, satisfiability in $T_{\Sigma/E}$ is decidable anyway, but *not* as conjectured.

A key reason for the failure of the above conjecture in Example 7 is that the rules in $R(E)$ *rewrite constructor terms*, so that not all constructor terms are in normal form. Therefore, the conjecture's mistake was to focus on $T_{\Sigma/B}$, when we should focus on the *canonical algebra of constructors* $C_{\mathcal{R}_\Omega}$ associated to a constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ of the given FVP decomposition $\mathcal{R} = (\Sigma, B, R)$. Note that the canonical term algebra $C_{\mathcal{R}}$ and the canonical constructor algebra $C_{\mathcal{R}_\Omega}$ are related by the equality $C_{\mathcal{R}}|_{\Omega} = C_{\mathcal{R}_\Omega}$. This allows

us to *reduce* satisfiability in $C_{\mathcal{R}}$ to satisfiability in $C_{\mathcal{R}_\Omega}$. To do so, first recall the decomposition $\mathcal{R}^\wedge = (\Sigma^\wedge, B, R)$ of the theory $(\Sigma^\wedge, E)$ introduced in Section 3 and note that if $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ is a constructor decomposition of $\mathcal{R} = (\Sigma, B, R)$, then $\mathcal{R}^\wedge_\Omega = (\Omega^\wedge, B_\Omega, R_\Omega)$ is a constructor decomposition of $\mathcal{R}^\wedge = (\Sigma^\wedge, B, R)$, where $\Omega^\wedge$ is obtained by adding to $\Omega$ the new sorts and the logical symbols $\_ = \_$, $\_ \neq \_$, and $\_ \wedge \_$ in $\Sigma^\wedge$. Therefore, given a conjunction of $\Sigma$-literals $\phi$, a *constructor* $R, B$*-variant* $(\phi', \theta)$ of $\phi$ is an $R, B$-variant $(\phi', \theta)$ of $\phi$ such that $\phi'$ is a conjunction of $\Omega$-literals. Here is now the desired reduction:

**Theorem 6.** *(Descent Theorem). Let a decomposition $\mathcal{R} = (\Sigma, B, R)$ of an OS equational theory $(\Sigma, E)$ protect a constructor decomposition $\mathcal{R}_\Omega$ with equational theory $(\Omega, E_\Omega)$. Then, a QF $\Sigma$-conjunction of literals $\phi$ is satisfiable in $T_{\Sigma/E}$ iff there is a constructor variant $(\phi', \theta)$ of $\phi$ such that $\phi'$ is satisfiable in $T_{\Omega/E_\Omega}$.*

*Proof.* We can replace $T_{\Sigma/E}$ by its isomorphic $C_{\mathcal{R}}$, and $T_{\Omega/E_\Omega}$ by its isomorphic $C_{\mathcal{R}_\Omega}$. Furthermore, by Definition 4, as $S$-sorted *sets* $C_{\mathcal{R}_\Omega} = C_{\mathcal{R}}$, and as $\Omega$-algebras, the unique isomorphism $h : C_{\mathcal{R}_\Omega} \cong C_{\mathcal{R}}|_\Omega$ is the identity function.

To prove the ($\Leftarrow$) implication, let $\phi$ be a conjunction of $\Sigma$-literals with variables $\overline{x}$, and $(\phi', \theta)$ a constructor variant of $\phi$ with variables $\overline{y}$, and therefore an $\Omega$-formula. $\phi'$ is satisfiable in $T_{\Omega/E_\Omega}$ iff it is satisfiable in $C_{\mathcal{R}_\Omega}$, say by an assignment $[\alpha] \in [\overline{y} \rightarrow C_{\mathcal{R}_\Omega}]$. But then we have the equivalences:

$$C_{\mathcal{R}_\Omega}, [\alpha]_{B_\Omega} \models \phi' \;\Leftrightarrow\; C_{\mathcal{R}}|_\Omega, [\alpha]_{B_\Omega} \models \phi' \;\Leftrightarrow\; C_{\mathcal{R}}, [(\theta\alpha)!_{R,B}]_{B_\Omega} \models \phi,$$

proving that $\phi$ is satisfiable in $T_{\Sigma/E}$, where for each variable $x$, $x(\theta\alpha)!_{R,B} = (x\theta\alpha)!_{R,B}$.

To prove the ($\Rightarrow$) implication note that for $\phi$ of the form $u_1 = v_1 \wedge \ldots \wedge u_n = v_n \wedge u'_1 \neq v'_1 \wedge \ldots \wedge u'_m \neq v'_m$ and with variables $\overline{x}$, an assignment $[\beta]_{B_\Omega} \in [\overline{x} \rightarrow C_{\mathcal{R}}]$ is such that $C_{\mathcal{R}}, [\beta]_{B_\Omega} \models \phi$ iff $(u_i\beta)!_{R,B} =_{B_\Omega} (v_i\beta)!_{R,B}$, $1 \leqslant i \leqslant n$, and $(u'_j\beta)!_{R,B} \neq_{B_\Omega} (v'_j\beta)_{R,B}$, $1 \leqslant j \leqslant m$. But by sufficient completeness of $\Omega$ with respect to $\mathcal{R}$ all the $(u_i\beta)!_{R,B}$, $(v_i\beta)!_{R,B}$, $(u'_j\beta)!_{R,B}$, and $(v'_j\beta)_{R,B}$ are $\Omega$-terms. Therefore, $((\phi\beta)!_{R,B}, \beta)$ is a *ground* $R, B$-variant of $\phi$, and therefore a constructor $R, B$-variant of $\phi$, valid, and therefore satisfiable, in $T_{\Omega/E_\Omega}$. □

This theorem has a useful corollary for equational OS-FO theories:

**Corollary 1.** *(Descent to Decidability). Let an FVP decomposition $\mathcal{R} = (\Sigma \cup \Pi, B, R)$ of an OS-FO equational theory $((\Sigma, \Pi), \Gamma)$, with $B$ having a finitary unification algorithm, protect a constructor decomposition $\mathcal{R}_{(\Omega,\Delta)} = (\Omega \cup \Delta, B_\Omega, R_{(\Omega,\Delta)})$ of a theory $((\Omega, \Delta), \Gamma_0)$, with $=_{B_\Omega}$ decidable and such that satisfiability of QF $(\Omega, \Delta)$-formulas in $T_{\Omega,\Delta,\Gamma_0}$ is decidable. Then, satisfiability of any QF $(\Sigma, \Pi)$-formula $\phi$ in $T_{\Sigma,\Pi,\Gamma}$ is decidable.*

*Proof.* By putting formulas in DNF we can reduce to the case where the QF formula $\phi$ is a conjunction of literals. By the Descent Teorem 6 a $(\Sigma, \Pi)$-conjunction $\phi$ is satisfiable in $T_{\Sigma,\Pi,\Gamma}$ iff $\widetilde{\phi}$ has a constructor $R, B$-variant $(\widetilde{\phi}', \theta)$ such that $\phi'$ is satisfiable in $T_{\Omega,\Delta,\Gamma_0}$. But by completeness of $[\![\widetilde{\phi}]\!]^{\Omega^\wedge}_{R,B}$, then there is a constructor $R, B$-variant $(\widetilde{\phi}'', \rho) \in [\![\widetilde{\phi}]\!]^{\Omega^\wedge}_{R,B}$ and a substitution $\gamma$ such that $\phi' =_B \phi''\gamma$.

Therefore, by composing with $\gamma$ a satisfying assignment for $\phi'$ we get a satisfying assignment for $\phi''$. But $[\![\widetilde{\phi}]\!]^{\Omega^{\wedge}}_{R,B}$ is an effectively computable finite set, and the satisfiability of $\phi''$ in $T_{\Omega,\Delta,\Gamma_0}$ for each $(\widetilde{\phi''},\rho) \in [\![\widetilde{\phi}]\!]^{\Omega^{\wedge}}_{R,B}$ is decidable by hypothesis. Therefore, we can decide the satisfiability of $\phi$ in $T_{\Sigma,\Pi,\Gamma}$ by testing whether $[\![\phi]\!]^{\Omega^{\wedge}}_{R,B}$ contains a formula $\widetilde{\phi''}$ with $\phi''$ satisfiable in $T_{\Omega,\Delta,\Gamma_0}$. □

Corollary 1 can be "unpacked" into an actual *generic algorithm* to decide the satisfiability in $T_{\Sigma,\Pi,\Gamma}$ of any QF $(\Sigma,\Pi)$-formula $\phi$ under those assumptions. We can first of all shift the problem to the equivalent one of satisfiability of the equational version $\widetilde{\phi}$ in $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}$ and, by assuming $\widetilde{\phi}$ in DNF,[4] we can reduce to deciding whether some conjunction of literals $\bigwedge G \wedge \bigwedge D$, with $G$ equations and $D$ disequations in such a DNF is satisfiable. The algorithm is as follows:

1. By the FVP assumption, $\widetilde{\Gamma}$-unification is finitary. Furthermore, thanks to Theorem 5, to test the satisfiability of $\bigwedge G$ we need only compute the finite set $Unif^{\Omega\cup\Delta}_{\widetilde{\Gamma}}(\bigwedge G)$ of its *constructor* $\widetilde{\Gamma}$-unifiers. In this way we can reduce to the case of deciding the satisfiability of some conjunction of disequalities $(\bigwedge D\alpha)!_{R,B}$, for some constructor unifier $\alpha \in Unif^{\Omega\cup\Delta}_{\widetilde{\Gamma}}(\bigwedge G)$.
2. For each such $(\bigwedge D\alpha)!_{R,B}$ we can then compute a finite, complete set of most general constructor $R,B$-variants $[\![(\bigwedge D\alpha)!_{R,B}]\!]^{\Omega\cup\Delta}_{R,B}$ to obtain a finite set of conjunctions of $\Omega\cup\Delta$-disequalities of the form $\bigwedge D'$.
3. We can then decide the satisfiability in $T_{\Omega\cup\Delta,\widetilde{\Gamma}_0}$ of each such $\bigwedge D'$, so that $\bigwedge G \wedge \bigwedge D$ will be satisfiable in $T_{\Sigma,\Pi,\Gamma}$ iff some $\bigwedge D'$ is so in $T_{\Omega\cup\Delta,\widetilde{\Gamma}_0}$.

In a sequential implementation of such an algorithm, steps (1) and (2) should be computed *incrementally*: one unifier, resp. variant, at a time. Maude 2.7 supports incremental computation of variants and variant-based unifiers with caching to reduce the cost of computing the next variant, resp. unifier, and such incrementality can be exploited in the computation of $Unif^{\Omega\cup\Delta}_{\widetilde{\Gamma}}(\bigwedge G)$ and $[\![(\bigwedge D\alpha)!_{R,B}]\!]^{\Omega\cup\Delta}_{R,B}$.

Note that a key property of this generic algorithm is that, thanks to the use of constructor unifiers and constructor variants, everything is reduced to checking the satisfiability of conjunctions of $\Omega\cup\Delta$-*disequalities* in $T_{\Omega\cup\Delta,\widetilde{\Gamma}_0}$.

The simplest case in which the above algorithm can be exploited is for $\mathcal{R} = (\Sigma, B, R)$ FVP with a finitary $B$-unification algorithm, $\Omega$ a signature of free constructors modulo $B_\Omega$, and satisfiability of QF formulas in $T_{\Omega/B_\Omega}$ decidable. In Section 6 I study such decidability for the commonly occurring case when $B_\Omega$ is any (possibly empty) combination of commutativity, associativity-commutativity, and identity axioms for some binary function symbols in $\Sigma$. Exploiting the descent theorem when $R_\Omega \neq \varnothing$ is postponed until Section 7.

[4] Using a lazy $DPLL(T)$ solver (see, e.g., [13]) we do *not* have to assume that $\varphi$ is in DNF: the $DPLL(T)$ solver will efficiently extract from $\varphi$ the appropriate conjunctions of $T$-literals to check for satisfiability.

## 6 OS-Compact Theories and Satisfiability in $T_{\Omega/ACCU}$

The simplest application of Corollary 1 and its associated generic algorithm is when $\mathcal{R} = (\Sigma, B, R)$ is FVP with a finitary $B$-unification algorithm, $\Omega$ is a signature of free constructors modulo $B_\Omega$, and satisfiability of QF formulas in $T_{\Omega/B_\Omega}$ is decidable. Generalizing a similar result in [30] for the unsorted and $AC$ case, I show below that, when $B_\Omega = ACCU$ —where $ACCU$ stands for any combination of associativity-commutativity ($AC$), commutativity ($C$), and/or left- or right-identity ($U$) axioms for some binary function symbols— satisfiability of QF formulas in $T_{\Omega/ACCU}$ is decidable. But, generalizing again another result in [30], we can view such a satisfiability result as part of a broader one, namely, decidable satisfiability in $T_{\Sigma,\Pi,\Gamma}$ or, equivalently, in $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}$ when $((\Sigma,\Pi),\Gamma)$ is an OS-*compact* equational OS-FO theory.

We first need some technical notions. Given an OS equational theory $(\Sigma, E)$, call a $\Sigma$-equality $u = v$ *$E$-trivial* iff $u =_E v$, and a $\Sigma$-disequality $u \neq v$ *$E$-consistent* iff $u \neq_E v$. Likewise, call a conjunction $\bigwedge D$ of $\Sigma$-disequalities *$E$-consistent* iff each $u \neq v$ in $D$ is so. Call a sort $s \in S$ *finite* in both $(\Sigma, E)$ and $T_{\Sigma/E}$ iff $T_{\Sigma/E,s}$ is a finite set, and *infinite* otherwise. Here is the key concept:

**Definition 7.** *An equational OS-FO theory $((\Sigma,\Pi),\Gamma)$ is called* OS-compact *iff: (i) for each sort $s$ in $\Sigma$ we can effectively determine whether $s$ is finite or infinite in $T_{\Sigma\cup\Pi/\widetilde{\Gamma},}$, and, if finite, can effectively compute a representative ground term $rep([u]) \in [u]$ for each $[u] \in T_{\Sigma\cup\Pi/\widetilde{\Gamma},s}$; (ii) $=_{\widetilde{\Gamma}}$ is decidable and $\widetilde{\Gamma}$ has a finitary unification algorithm; and (iii) any finite conjunction $\bigwedge D$ of negated $(\Sigma,\Pi)$-atoms whose variables have all infinite sorts and such that $\bigwedge \widetilde{D}$ is $\widetilde{\Gamma}$-consistent is satisfiable in $T_{\Sigma,\Pi,\Gamma}$.*

*We call an OS equational theory $(\Sigma, E)$ OS-*compact *iff the OS-FO theory $((\Sigma,\varnothing), E)$ is so.*

Note that this generalizes the notion of *compact theory* in [30] in four ways: (i) from unsorted to OS theories; (ii) by dealing with the phenomenon of possibly having some sorts finite and some infinite; (iii) by extending the notion from equational theories to OS-FO equational theories; and (iv) by including the case of computable *finite initial models*, because an OS-FO theory $((\Sigma,\varnothing), E)$ whose sorts are all finite and for which we can effectively compute representatives has decidable equality and finitary unification, and is OS-compact in a vacuous sort of way; e.g., the Boolean theory $\mathcal{B}$ of Example 1, and the theory $\mathcal{N}_2$ in Example 7 are both OS-compact. I will illustrate with examples that extensions (i)–(iii) are needed in many useful applications.

The key theorem about OS-compact theories is again a generalization of a similar one in [30]. I include its short proof to make the paper self-contained.

**Theorem 7.** *Let $((\Sigma,\Pi),\Gamma)$ be an* OS-compact *theory. The satisfiability of QF $(\Sigma,\Pi)$-formulas in $T_{\Sigma,\Pi,\Gamma}$ is decidable.*

*Proof.* Since $T_{\Sigma,\Pi,\Gamma}, \alpha \models \phi$ iff $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}, \alpha \models \widetilde{\phi}$, we can equivalently prove that for any QF formula $\phi$ its equational version $\widetilde{\phi}$ is decidable in $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}$. Assuming $\widetilde{\phi}$ in DNF, $\widetilde{\phi}$ will be satisfiable iff one of the conjunctions of atoms in the disjunction is satisfiable. Let us consider one such conjunction $\bigwedge G \wedge \bigwedge D$, with $G$ equations and $D$ disequations. $\bigwedge G \wedge \bigwedge D$ is satisfiable in $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}$ iff $\bigvee_{\alpha\in Unif_{\widetilde{\Gamma}}(\bigwedge G)} \bigwedge D\alpha$ is so. Consider now any of the $\bigwedge D\alpha$, and let $\overline{x}$, resp., $\overline{y}$, be its variables with finite (resp. infinite) sort. $\bigwedge D\alpha$ is satisfiable in $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}$ iff $\bigvee_{\beta\in[\overline{x}\rightarrow T_{\Sigma\cup\Pi/\widetilde{\Gamma}}]} \bigwedge D\alpha\, rep(\beta)$ is so, where, for each $x \in \overline{x}$, $rep(\beta)(x) = rep(\beta(x))$. But the variables of any such $\bigwedge D\alpha\, rep(\beta)$ are $\overline{y}$ and, having infinite sorts, the satisfiability of $\bigwedge D\alpha\, rep(\beta)$ in $T_{\Sigma\cup\Pi/\widetilde{\Gamma}}$, and therefore that of $\widetilde{\phi}$, is decidable, as desired. □

This now gives us the following, quite useful corollary of Corollary 1:

**Corollary 2.** *Let an FVP decomposition $\mathcal{R} = (\Sigma \cup \Pi, B, R)$ of an OS-FO equational theory $((\Sigma, \Pi), \Gamma)$, with $B$ having a finitary unification algorithm, protect a constructor decomposition $\mathcal{R}_{(\Omega,\Delta)} = (\Omega \cup \Delta, B_\Omega, R_{(\Omega,\Delta)})$ of an OS-compact theory $((\Omega, \Delta), \Gamma_0)$, with $=_{B_\Omega}$ decidable. Then, satisfiability of any QF $(\Sigma, \Pi)$-formula $\phi$ in $T_{\Sigma,\Pi,\Gamma}$ is decidable.*

This corollary further "unpacks" how the satisfiability in $T_{\Omega,\Delta,\Gamma_0}$ of an $\Omega\cup\Delta$-disjunction of disequalities $\bigwedge D'$ obtained in step (2) of the generic satisfiability decision procedure "unpacking" Corollary 1 can be checked in step (3) when $((\Omega, \Delta), \Gamma_0)$ is OS-compact, namely, we then replace $\bigwedge D'$ by the disjunction of all the representative ground instantiations $\bigwedge D' rep(\beta)$ of its finite sort variables, and then check whether at least one such $\bigwedge D' rep(\beta)$ is satisfiable by checking the $B_\Omega$-consistency of $(\bigwedge D' rep(\beta))!_{R_{(\Omega,\Delta)},B_\Omega}$.

### 6.1 Theories $(\Omega, ACCU)$ are OS-Compact

Consider now an OS signature $\Omega$ where some (possibly empty) subsignature $\Omega_{ACCU} \subseteq \Omega$ of binary operators of the form $f : s\, s \rightarrow s$, for some $s \in S$, satisfy any combination of: (i) the associativity-commutativity ($AC$) axioms $f(f(x, y), z) = f(x, f(y, z))$ and $f(x, y) = f(y, x)$; (ii) just the commutativity ($C$) axiom $f(x, y) = f(y, x)$; (iii) the left-unit ($LU$) axiom $f(e_f, x) = x$ for a unit constant $e_f$; or (iv) the right-unit ($RU$) axiom $f(x, e_f) = x$ (note that the standard unit axioms ($U$) are just the combination of $LU$ and $RU$). Furthermore, if $f : s\, s \rightarrow s \in \Omega_{ACCU}$ belongs to a subsort polymorphic family $f^{[s]\,[s]}_{[s]}$, then all other members of the family are of the form $f : s'\, s' \rightarrow s'$, $f^{[s]\,[s]}_{[s]} \subseteq \Omega_{ACCU}$, and all operators in such a family satisfy exactly the same axioms. $ACCU$ abbreviates: *any combination of associativity-commutativity and/or commutativity and/or unit axioms*. Since all the above axiom combinations are possible and $\Omega_{ACCU}$ can be empty, the acronym $ACCU$, covers in fact *eight* possibilities for each subsort polymorphic family $f^{[s]\,[s]}_{[s]}$ of binary function symbols: (i) the "free"

case where $f$ satisfies no axioms; (ii) the case where $f$ is only $LU$; (iii) the case where $f$ is only $RU$; (iv) the case where $f$ is only $U$; (v) the case where $f$ is $C$; (vi) the case where $f$ is $CU$; (vii) the case where $f$ is $AC$; and (viii) the case where $f$ is $ACU$. Furthermore, I will always assume that $\Omega$ is $ACCU$-preregular.[5]

The main goal of this section is to prove that, under the above assumptions, satisfiability of QF $\Omega$-formulas in $T_{\Omega/ACCU}$ is decidable. This result generalizes from the unsorted to the order-sorted case, and from $AC$ to $ACCU$ axioms, a previous result by H. Comon-Lundh [30]. This is done in Theorem 8 below. But we need before the following auxiliary proposition, generalizing to the order-sorted and $ACCU$ case a similar result in [30] for the unsorted and $AC$ case:

**Proposition 1.** *Under the above assumptions, let $u = v$ be an $ACCU$-non-trivial $\Omega$-equation whose only variable is $x : s$. Then the set of most general $ACCU$-unifiers $Unif_{ACCU}(u = v)$ is finite, and all unifiers in it are ground unifiers, i.e., ground substitution $\{x : s \mapsto w\}$, with $w \in T_{\Omega,s}$. Since ground unifiers cannot be further instantiated, for any $ACCU$-unifier $\alpha$ there is a $\beta \in Unif_{ACCU}(u = v)$ with $\alpha =_{ACCU} \beta$.*

Since the proof is an inductive proof involving a somewhat lengthy case analysis, it is exiled to Appendix A. Note that for arbitrary combinations of associativity $A$, commutativity $C$, and left $LU$, and right $RU$ unit axioms, the above proposition is *as general as possible*: any combination of axioms involving associativity without commutativity will violate the requirement that $Unif_{ACCU}(u = v)$ is finite. Not only is it well-known that $A$ and $AU$ unification are in general infinitary: this also remains true when $u = v$ has a single variable $x$. For example, if $\_\cdot\_$ is an $A$ operator, and $a$ a constant, the equation $a \cdot x = x \cdot a$ has an infinite number of ground $A$ unifiers: $\{x \mapsto a\}$, $\{x \mapsto a \cdot a\}$, $\{x \mapsto a \cdot a \cdot a\}$, and so on.

The following theorem generalizes an analogous one in [30] for the unsorted and $AC$ case. Since the proof is quite short, and its argument makes fewer requirements on the reader than the corresponding one in [30], I include it to make the paper self-contained.

**Theorem 8.** *Under the above assumptions, satisfiability of QF $\Omega$-formulas in $T_{\Omega/ACCU}$ is decidable.*

*Proof.* By Theorem 7 it is enough to prove that $(\Omega, ACCU)$ is OS-compact. A proof that finiteness of sorts in $T_{\Omega/ACCU}$ is decidable and that for each finite sorts $s$ we can effectively describe its equivalence classes and choose representatives is given in Appendix B. Of course, order-sorted $ACCU$-unification (possibly extended with free function symbols) is finitary.[6]

We have to prove that if the sorts of all variables in $D$ are infinite and $\bigwedge D$ is $ACC$-consistent, then it is satisfiable in $T_{\Omega/ACC}$. The proof is by induction

[5] In the more relaxed sense that $\Omega$ is $B$-preregular for $B$ the non-unit $ACCU$ axioms and the unit axioms are *sort-decreasing* when oriented as left-to-right rewrite rules.

[6] See the discussion at the beginning of the proof of Proposition 1 in Appendix A.

on the number $n$ of variables in $D$. If $n = 0$, the result follows trivially. Otherwise, assume that the result holds for finite conjunctions of disequalities with $n$ variables, let $D$ have variables $x_1{:}s_1, \ldots, x_{n+1}{:}s_{n+1}$, and consider $D$ on the signature $\Omega \cup \{x_1 : s_1, \ldots, x_n : s_n\}$, where we have added the first $n$ variables as *new constants*, so that, on $\Omega \cup \{x_1 : s_1, \ldots, x_n : s_n\}$, $D$ has $x_{n+1} : s_{n+1}$ as its only variable. By Proposition 1, for each $u \neq v \in D$ having $x_{n+1}{:}s_{n+1}$ as a variable, there is a finite number of ground solutions for $x_{n+1} : s_{n+1}$ in the extended signature. But, since $T_{\Omega/ACCU,s_{n+1}}$ is infinite, and $T_{\Omega/ACCU,s_{n+1}} \subseteq T_{\Omega \cup \{x_1:s_1,\ldots,x_n:s_n\}/ACCU,s_{n+1}}$, we can choose a $w \in T_{\Omega/ACCU,s_{n+1}}$ that is different modulo $ACCU$ from any such solution for any such $u \neq v \in D$, so that for each $u \neq v \in D$ we have $u\{x_{n+1}{:}s_{n+1} \mapsto w\} \neq_{ACCU} v\{x_{n+1}{:}s_{n+1} \mapsto w\}$. Therefore, the induction hypothesis applies to $\bigwedge D\{x_{n+1}{:}s_{n+1} \mapsto w\}$, so that there is a satisfying assignment $[\theta]_{ACCU} \in [\{x_1{:}s_1, \ldots, x_n{:}s_n\} \to T_{\Omega/ACCU}]$ such that $T_{\Omega/ACCU}, [\theta] \models \bigwedge D\{x_{n+1}{:}s_{n+1} \mapsto w\}$, and therefore, since $w \in T_{\Omega,s_{n+1}}$, $T_{\Omega/ACCU}, [\theta \uplus \{x_{n+1}{:}s_{n+1} \mapsto w\}] \models \bigwedge D$, proving compactness and therefore the theorem. □

The above theorem yields as a direct consequence the decidable satisfiability of any QF equational formula in the the *natural numbers with addition.*

*Example 8.* (Natural Numbers with +). This is an unsorted[7] theory theory $\mathcal{N}_+^u$ with single sort *Nat*. The operations in the signature $\Omega$ are: $0 :\to Nat$, $1 :\to Nat$, and $\_ + \_ : Nat\ Nat \to Nat$, which satisfies the $ACU$ axioms, with 0 as unit. Since the conditions in Theorem 8 are met, satisfiability (and therefore validity) in the initial algebra of $\mathcal{N}_+^u$ is decidable.

By Theorem 7, deciding satisfiability of a conjunction $\bigwedge G \wedge \bigwedge D$ in the initial algebra of $\mathcal{N}_+^u$ boils down to computing the most general $ACU$-unifiers $\alpha$ of $\bigwedge G$, and then checking the $ACU$-consistency of each $\bigwedge D\alpha$, i.e., checking for each $u\alpha \neq v\alpha$ in $D\alpha$ that $u\alpha \neq_{ACU} v\alpha$. Note that $ACU$-unification is NP-complete [63].

For example, $n = 0 \vee n + n \neq n$ is a theorem in the initial algebra of $\mathcal{N}_+^u$ because its negation $n \neq 0 \wedge n + n = n$ is such that $n + n = n$ has $\{n \mapsto 0\}$ as its only $ACU$-unifier, yielding the unsatisfiable disequality $0 \neq 0$.

### 6.2 The Descent Theorem with Free Constructors Modulo *ACCU*

Thanks to Theorem 8 we can apply Corollary 2 to the case of an FVP decomposition $\mathcal{R} = (\Sigma, B, R)$ of an equational theory $(\Sigma, E)$, with $B$ having a finitary unification algorithm, and protecting the constructor decomposition $\mathcal{R}_\Omega = (\Omega, ACCU, \varnothing)$ of $(\Omega, ACCU)$ to obtain a method to decide the satisfiability of any QF $\Sigma$-formula in $T_{\Sigma/E}$. Let us see some examples.

[7] An order-sorted version $\mathcal{N}_+$ of $\mathcal{N}_+^u$ is obtained by adding a subsort inclusion $NzNat < Nat$, where *NzNat* denotes the non-zero naturals, typing 1 with sort *NzNat*, and adding the typing $\_ + \_ : NzNat\ NzNat \to NzNat$. $\mathcal{N}_+$ is also OS-compact for the exact same reasons. See Section 8 for a reduction of satisfiability in the initial agebra of $\mathcal{N}_+$ to satisfiability in the initial algebra of $\mathcal{N}_+^u$. $\mathcal{N}_+$ makes the language more expressive: instead of stating $x \neq 0$ we can just type $x$ as having sort *NzNat*.

*Example 9.* (Naturals with *zero*? Predicate). Recall the FVP decomposition $\mathcal{N}_{zero?}$ from Example 2. Since $\Omega = \Sigma - \{zero?\}$ is a signature of free constructors, the conditions of Corollary 2 are met. Let now $\phi$ be the formula $x = zero?(n) \ \wedge \ x \neq \top \ \wedge \ x \neq \bot$. Recall that its two constructor variants are $x = \bot \ \wedge \ x \neq \top \ \wedge \ x \neq \bot$, and $x = \top \ \wedge \ x \neq \top \ \wedge \ x \neq \bot$. Solving the equation in each case we get formulas $\bot \neq \top \ \wedge \ \bot \neq \bot$, and $\top \neq \top \ \wedge \ \top \neq \bot$, which have both $\varnothing$-inconsistent disequalities, so $\phi$ is unsatisfiable in $C_{\mathcal{N}_{zero?}}$. This solves the "puzzle" in Example 2.

*Example 10.* (Naturals Modulo 2). Recall the theory $\mathcal{N}_2^{os} = (\Sigma, \varnothing, R(E))$ from Example 6, with a free constructor decomposition $(\Omega, \varnothing, \varnothing)$. Therefore, Corollary 2 applies and QF satisfiability in the initial algebra of $\mathcal{N}_2^{os}$ is decidable. This is of course obvious, since $C_{\mathcal{N}_2^{os}}$ is a finite, computable algebra, so that $\mathcal{N}_2^{os}$ itself is already OS-compact as pointed out after Definition 7. But since decidable QF satisfiability on $C_{\mathcal{N}_2^{os}}$ also follows from the general method of descent to compact theories, it may still be useful to illustrate this general method. All sorts are finite. The only constructor terms of sort *Zero*, resp. *One*, are 0, resp. $s(0)$, and the only constructor terms in the top sort *Nat* are 0 and $s(0)$. The disjunction $x = 0 \ \vee \ x = s(0)$ is a theorem of $C_{\mathcal{N}_2^{os}}$ because its negation $x \neq 0 \wedge x \neq s(0)$ is unsatisfiable. Recall that in the finite sort case this is settled by instantiation: the instantiations $0 \neq 0 \wedge 0 \neq s(0)$ and $s(0) \neq 0 \wedge s(0) \neq s(0)$ are both unsatisfiable. This solves the "puzzle" in Example 7.

This example could be extended in various ways. We could: (i) likewise define $\mathcal{N}_n^{os} = (\Sigma, \varnothing, R(E))$ for any $n \geqslant 1$, and (ii) add addition, multiplication, or other operations (e.g., $n$-valued logic connectives) to $\mathcal{N}_n^{os}$ for any $n \geqslant 1$. The example points outs a weakness in the treatment of finite sorts: with many variables, even if the instantiation of variables by concrete values is done incrementally, satisfiability checking will become very inefficient: an encoding into SAT or bit vectors should perform much better.

*Example 11.* (Natural Presburger Arithmetic). $\mathcal{N}_+^u$ in Example 8 can be extended to an FVP decomposition $\mathcal{N}_{+,>^b,\geqslant^b}^u$ having the natural numbers with $+$ and with the order relations $>$ and $\geqslant$ as a Boolean-valued predicates[8] as follows. We add a new sort *Truth* with constants $\bot$ and $\top$, and two defined functions $\_ > \_, \_ \geqslant \_ : Nat\ Nat \rightarrow Truth$ with rules $1 + m + n > n \rightarrow \top$, $m > m + n \rightarrow \bot$ and $m + n \geqslant m \rightarrow \top$ and $n \geqslant 1 + m + n \rightarrow \bot$. This specification is sufficiently complete with $\mathcal{N}_+^u$ extended with $\top, \bot$ as its constructor subspecification, and yields an FVP decomposition with variant complexity 6.

Since $\mathcal{N}_+^u$ extended with $\top, \bot$ is OS-compact, by Corollary 2 satisfiability in the initial algebra of $\mathcal{N}_{+,>^b,\geqslant^b}^u$ is decidable. For example, the transitivity law $(n > m = \top \wedge m > k = \top) \Rightarrow n > k \neq \bot$ of natural Presburger arithmetic is a theorem because its negation is the conjunction $n > m = \top \wedge m > k = \top \wedge n > k = \bot$, which has no variant-based unifiers. Likewise, the equivalence

[8] See Example 22 for a version $\mathcal{N}_{+,>,\geqslant}$ of natural Pressburger arithmetic in which $>$ and $\geqslant$ are only explictly defined in the positive case.

$x \geqslant y \Leftrightarrow \neg(y > x)$ is a theorem of natural Presburger arithmetic because it can be expressed as the equational formula $(x \geqslant y \neq \top \vee y > x \neq \top) \wedge (y > x \neq \bot \vee x \geqslant y \neq \bot)$, which when negated gives us the disjunction of conjunctions $(x \geqslant y = \top \wedge y > x = \top) \vee (y > x = \bot \wedge x \geqslant y = \bot)$, each of which having no variant-based unifiers.

The fact that $\mathcal{N}^u_{+,>^b,\geqslant^b}$ has the finite sort *Truth* might seem to raise the concern of a combinatorial explosion due to the instantiation by constants of variables of sort *Truth*, as already discussed for $\mathcal{N}^{os}_2$ in Example 10. But this is an optical illusion: the non-equality atoms of Presburger arithmetic are of the form $u > v$ or $u \geqslant v$, which are here represented by the equality atoms $u > v = \top$ or $u \geqslant v = \top$. Therefore, the equational version $\widetilde{\varphi}$ of a Presburger arithmetic formula has *no* variables of sort *Truth*, so the problem never arises.

*Example 12.* (Natural Numbers with $+$, *max*, *min* and $\dot{-}$). This is the decomposition $\mathcal{N}^u_{+,max,min,\dot{-}}$ obtained by adding to $\mathcal{N}^u_+$ the max and min operators $max, min : Nat\ Nat \to Nat$ and the "monus" operator $\_\dot{-}\_ : Nat\ Nat \to Nat$ as defined functions, and the commutativity axioms for *max* and *min*, which are respectively defined by the rules $max(n, n+m) \to n+m$, $min(n, n+m) \to n$. Likewise, $\dot{-}$ is defined by the rules, $n \dot{-} (n+m) \to 0$, $(n+m) \dot{-} n \to m$. $\mathcal{N}^u_{+,max,min,\dot{-}}$ is FVP with variant complexity 9. Furthermore, $\mathcal{N}^u_{+,max,min,\dot{-}}$ protects the OS-compact constructor decomposition $\mathcal{N}^u_+$ so that, by Corollary 2, satisfiability (and therefore validity) in $C_{\mathcal{N}^u_{+,max,min,\dot{-}}}$ is decidable. For example, $n \dot{-} m = 0 \vee m \dot{-} n = 0$ is a theorem in $C_{\mathcal{N}^u_{+,max,min,\dot{-}}}$, because its negation $n \dot{-} m \neq 0 \wedge m \dot{-} n \neq 0$ has constructor variants $0 \neq 0 \wedge m \neq 0$ and $0 \neq 0 \wedge m' \neq 0$ (obtained with two different substitutions), both *ACU*-inconsistent.

*Example 13.* (Integers Offsets). This is perhaps the simplest possible theory $\mathcal{Z}_{s,p}$ of integers. Decisions procedures for it have been given in [22,18,4]. This example is also interesting because it is usually specified in an unsorted way, making it *impossible* to have a signature of free constructors. Instead, an order-sorted specification has a signature of free constructors, allowing Corollary 2 to be applied. The sorts are: *Int*, *Nat*, *Neg*, and *Zero*, with subsort inclusions $Zero < Nat\ Neg < Int$. The subsignature $\Omega$ of free constructors is $0 :\to Zero$, $s : Nat \to Nat$, and $p : Neg \to Neg$, and the defined symbols[9] $s, p : Int \to Int$. The rules $R$ are just $p(s(m)) \to m$ and $s(p(n)) \to n$, with $m$ of sort *Nat* and $n$ of sort *Neg*.

Since $\mathcal{Z}_{s,p}$ is FVP with variant complexity 4 and is sufficiently complete with signature of free constructors $\Omega$, the conditions of Corollary 2 are met and satisfiability, and therefore validity, in $C_{\mathcal{Z}_{s,p}}$ is decidable. Let us, for example, decide the validity of the inductive theorem $s(x) = s(y) \Rightarrow x = y$, with $x, y$ of sort *Int*. This is equivalent to checking that $s(x) = s(y) \wedge x \neq y$ is unsatisfiable. The only variant-based $E$-unifier of $s(x) = s(y)$, $\{x \mapsto y\}$, yields the inconsistent disequality $y \neq y$. Thus, $s(x) = s(y) \Rightarrow x = y$ holds in $C_{\mathcal{Z}_{s,p}}$.

[9] Note the interesting phenomenon, impossible in a many-sorted setting, that subsort-polymorphic symbols like $s$ or $p$ can be constructors for some typings and a defined symbols for other typings. The same happened for $s$ in $\mathcal{N}^{os}_2$.

*Example 14.* (Integers with Addition). The decomposition $\mathcal{Z}_+$ for integers with addition imports in a protecting mode the theory $\mathcal{N}_+$ of natural numbers with addition in Footnote 7, and extends its constructor signature by adding two new sorts, *NzNeg*, and *Int*, with subsort inclusions *Nat NzNeg* $<$ *Int*, and a constructor $-$ : *NzNat* $\rightarrow$ *NzNeg*, to get an extended constructor signature $\Omega$. The only defined function symbol is: $\_ + \_$ : *Int Int* $\rightarrow$ *Int*, also *ACU*. The rewrite rules $R$ defining $+$ and making $(\Omega, ACU, \emptyset)$ an *ACU*-free constructor decomposition of $\mathcal{Z}_+$ are the following (with $i$ a variable of sort *Int*, and $n, m$ variables of sort *NzNat*): $i + n + -(n) \rightarrow i$, $i + -(n) + -(m) \rightarrow i + -(n + m)$, $i + n + -(n + m) \rightarrow i + -(m)$, and $i + n + m + -(n) \rightarrow i + m$. Note that, by the *ACU* axioms, the initial algebra $C_{\mathcal{Z}_+}$ is automatically a *commutative monoid*. Furthermore, by sufficient completeness $C_{\mathcal{Z}_+}|_\Omega = T_{\Omega/ACU}$, so that the first rule (specialized to $i = 0$) plus the $U$ axioms (specialized to $x = 0$) make $C_{\mathcal{Z}_+}$ into an *abelian group*, since it satisfies the axiom $(\forall x)(\exists y)\ x + y = 0$.

Subsorts make the language of $\mathcal{Z}_+$ considerably more expressive than an untyped language: we do not have to say $x > 0$ (resp. $x < 0$) by additionally defining an order predicate $>$: we just type $x$ with sort *NzNat* (resp. *NzNeg*).

$\mathcal{Z}_+$ is FVP with variant complexity 12. Since the conditions of Corollary 2 are met, satisfiability, and therefore validity, in $C_{\mathcal{Z}_+}$ is decidable. Let us, for example, decide the validity of the inductive theorem $i + j = i + l \Rightarrow j = l$, with $i, j, l$ variables of sort *Int*. This is equivalent to checking that $i + j = i + l \wedge j \neq l$ is unsatisfiable. The only variant unifier of $i + j = i + l$ is $\{j \mapsto l\}$, giving us $l \neq l$, which is *AC*-inconsistent.

*Example 15.* (Integer Presburger Arithmetic). The FVP theory $\mathcal{Z}_{+,>^b,\geqslant^b}$ of integer Presburger arithmetic with Boolean-valued[10] $>$ and $\geqslant$ protects $\mathcal{Z}_+$ by adding a new sort *Truth* with constants $\bot$ and $\top$, and defined functions $\_ > \_$, $\_ \geqslant \_$ : *Int Int* $\rightarrow$ *Truth* with rules $p + n > n \rightarrow \top$, $n > -(q) \rightarrow \top$, $-(p) > -(p + q) \rightarrow \top$, $n > n + m \rightarrow \bot$, $-(q) > n \rightarrow \bot$, $-(p) > -(p) \rightarrow \bot$, and $-(p + q) > -(p) \rightarrow \bot$ for $>$, and rules $n + m \geqslant n \rightarrow \top$, $n \geqslant -(q) \rightarrow \top$, $-(p) \geqslant -(p) \rightarrow \top$, $-(p) \geqslant -(p + q) \rightarrow \top$, $n \geqslant p + n \rightarrow \bot$, $-(q) \geqslant n \rightarrow \bot$, and $-(p + q) \geqslant -(p) \rightarrow \bot$ for $\geqslant$, were $p, q$ have sort *NzNat*, $n$ has sort *Nat*, and $i, j$ have sort *Int*. $\mathcal{Z}_{+,>^b,\geqslant^b}$ is sufficiently complete with constructor subspecification that of $\mathcal{Z}_+$ extended with $\top, \bot$, and FVP with variant complexity 28.

Since the constructor subspecification of $\mathcal{Z}_+$ extended with $\top, \bot$ is OS-compact, by Corollary 2 satisfiability in the initial algebra of $\mathcal{Z}_{+,>^b,\geqslant^b}$ is decidable. For example, the transitivity law $(i > j = \top \wedge j > k = \top) \Rightarrow i > k \neq \bot$ of integer Presburger arithmetic is a theorem because its negation is the conjunction $i > j = \top \wedge j > k = \top \wedge i > k = \bot$, which has no variant-based unifiers. Likewise, the equivalence $i \geqslant j \Leftrightarrow \neg(j > i)$ is a theorem of integer Presburger arithmetic because it can be expressed as the equational formula $(i \geqslant j \neq \top \vee j > i \neq \top) \wedge (j > i \neq \bot \vee i \geqslant j \neq \bot)$, which when negated gives us the disjunction of conjunctions $(i \geqslant j = \top \wedge j > i = \top) \vee (j > i = \bot \wedge i \geqslant j = \bot)$, each of which having no variant-based unifiers.

[10] See Example 23 for an even simpler version $\mathcal{Z}_{+,>,\geqslant}$ of integer Presburger arithmetic in which $>$ and $\geqslant$ are only explicitly defined in the positive case.

*Example 16.* (If-then-else). Let $\mathcal{R}$ be any FVP theory protecting the FVP theory $\mathcal{B}$ of Example 1 and having a constructor decomposition of the form $(\Omega, B, \varnothing)$ (this means that the only constructor operators of sort *Truth* are $\top$ and $\bot$). Let now $A$ be a sort in $\mathcal{R}$ which is the top sort of its connected component and different[11] from *Truth*. We then define the theory $\mathcal{R}_{if[A]}$ by adding to $\mathcal{R}$ a fresh new "if-then-else" operator $[\_,\_,\_] : \mathit{Truth}\ A\ A \to A$ and two rules: $[\top, x, y] \to x$ and $[\bot, x, y] \to y$, with $x, y$ of sort $A$. Then $\mathcal{R}_{if[A]}$ is also FVP, has also $(\Omega, B, \varnothing)$ as its constructor decomposition, and if $\mathcal{R}$ had variant complexity $k$, then $\mathcal{R}_{if[A]}$ has variant complexity $k+3$. Therefore, by Corollary 2 satisfiability in the initial algebra of $\mathcal{R}_{if[A]}$ is decidable.

For a simple instance of this general construction, let $\mathcal{N}_{+,>^b,\geqslant^b,bops}$ be the theory union $\mathcal{N}_{+,>^b,\geqslant^b} \cup \mathcal{B}$. Then $\mathcal{N}_{+,>^b,\geqslant^b,bops,if[Nat]}$ is just natural Presburger arithmetic enriched with Boolean operations and an if-then-else operator.

Validity of any QF formula in the initial algebra of $\mathcal{N}_{+,>^b,\geqslant^b,bops,if[Nat]}$ is decidable. For example, we can decide the validity of the "premise interchange law" $[b,[b',x,y],[b',x',y']] = [b',[b,x,x'],[b,y,y']]$, with $b, b'$ of sort *Truth* and $x, y, x', y'$ of sort *Nat* (see, e.g., [16,76] for various other laws). Negated, this becomes the disequality $[b,[b',x,y],[b',x',y']] \neq [b',[b,x,x'],[b,y,y']]$. Its constructor variants are the disequalities: $x \neq x$, $y \neq y$, $x' \neq x'$ and $y' \neq y'$, all inconsistent, so the negation of the law is unsatisfiable and the law valid.

## 7 Satisfiability in Parameterized FVP Data Types

What Corollary 2 achieves is a large increase in the infinite class of decidable OS-FO equational theories for which satisfiability of QF formulas in their initial models is decidable, namely, it grows from the class of OS-compact theories, including those of the form $(\Omega, ACCU)$, to that of all those OS-FO equational theories having an FVP theory decompositions with axioms $B$ having a finitary unification algorithm and protecting an OS-compact constructor subtheory.

But how can we further enlarge the class of OS-FO equational theories for which satisfiability of QF formulas in their initial model is decidable? Example 16 is suggestive in this regard, since it is not really a theory, but a *theory transformation* $\mathcal{R} \mapsto \mathcal{R}_{if[A]}$ mapping an FVP theory $\mathcal{R}$ with axioms $B$ having a finitary unification algorithm and protecting an OS-compact constructor subtheory into another FVP theory having the *same* axioms $B$ and OS-compact constructor subtheory, and therefore decidable satisfiability for its initial model.

This suggests the much more general idea of *parameterized data types*, which are theory transformations of this kind applicable to a typically infinite class of input theories and yielding an equally infinite class of instantiated theories.

[11] There is no real loss of generality in assuming $A \neq \mathit{Truth}$: we could drop this requirement, but an if-then-else operator $[\_,\_,\_]$ becomes unnecessary for $A = \mathit{Truth}$, since it is already available as the definitional extension $[x,y,z] = (x \wedge y) \vee (\neg(x) \wedge z)$. There is no real loss of generality either in singling out just one top sort $A$, since we can iterate the same construction over the top sorts of all connected components.

Therefore, an appealing idea is to search for *satisfiability-preserving* parameterized data types. That is, parameterized data types that, under suitable conditions, transform an input theory with decidable satisfiability of QF formulas in its initial model into a corresponding instance of the parameterized data type with the same property for *its* initial model.

I will give a full treatment of parameterized FVP data types elsewhere. Here, I illustrate with several examples a general method for substantially enlarging, by means of parameterization, the class of equational OS-FO theories with initial models having decidable QF satisfiability. For my present purposes it will be enough to summarize the basic general facts and assumptions for the case of FVP parameterized data types with a *single parameter* $X$. That is, I will restrict myself to parameterized FVP theories of the form $\mathcal{R}[X] = (\mathcal{R}, X)$, where $\mathcal{R} = (\Sigma \cup \Pi, B, R)$ is an FVP decomposition of a finitary equational OS-FO theory $((\Sigma, \Pi), \Gamma)$; and $X$ is a sort in $\Sigma$ (called the *parameter sort*) such that: (i) is empty,[12] i.e., $T_{\Sigma \cup \Pi / \tilde{\Gamma}, X} = \emptyset$; and (ii) $X$ is a minimal element in the sort order, i.e., there is no other sort $s'$ with $s' < X$.

Consider now an FVP decomposition $\mathcal{G} = (\Sigma' \cup \Pi', B', R')$ of another finitary OS-FO equational theory $((\Sigma', \Pi'), \Gamma')$, which we can assume without loss of generality[13] disjoint from $((\Sigma, \Pi), \Gamma)$, and let $s$ be a sort in $\Sigma'$. The *instantiation* $\mathcal{R}[\mathcal{G}, X \mapsto s] = (\Sigma[\Sigma', X \mapsto s], B \cup B', R \cup R')$ is the decomposition of a theory $(\Sigma[\Sigma', X \mapsto s], E \cup E')$, extending $(\Sigma', E')$, where the signature $\Sigma[\Sigma', X \mapsto s]$ is defined as the union $\Sigma[X \mapsto s] \cup \Sigma'$, with $\Sigma[X \mapsto s]$ just like $\Sigma$, except for $X$ renamed to $s$. The set of sorts is $(S - \{X\}) \uplus S'$, and the poset ordering is obtained by combining those of $\Sigma[X \mapsto s]$ and $\Sigma'$.

$\mathcal{R}[\mathcal{G}, X \mapsto s]$ is also FVP under fairly mild assumptions. The only problematic issue is termination, because the disjoint union of terminating rewrite theories need not be terminating [91]. However, many useful $p$-termination properties $p$ ensuring the $p$-termination of a disjoint union have been found (see, e.g., [56]). Therefore I will assume that either: (i) $\mathcal{R}[X]$ and $\mathcal{G}$ are both $p$-terminating for a modular termination property $p$, or (ii) $\mathcal{R}[\mathcal{G}, X \mapsto s]$ has been proved terminating. Convergence of $\mathcal{R}[\mathcal{G}, X \mapsto s]$ then follows easily from termination, because there are no new critical pairs. So does the FVP property, which is a modular property (see, e.g., [19]). In fact one can say more: the variant complexity of $\mathcal{R}[\mathcal{G}, X \mapsto s]$ is the *sum* of those of $\mathcal{R}[X]$ and $\mathcal{G}$. We furthermore require the *parameter protection* property that the unique $\Sigma'$ homomorphism $h : T_{\Sigma'/E'} \to T_{\Sigma[\Sigma', X \mapsto s]/E \cup E'}|_{\Sigma'}$ is an isomorphism. Typically, parameter protection can be easily proved using a protected constructor subtheory $\mathcal{R}_{(\Omega, \Delta)}[X]$.

[12] This violates the general assumption that sorts are non-empty; however, parameter sort instantiations to target theories with non-empty sorts make $X$ non-empty.

[13] There is no real loss of generality because we can make it so by renaming its sorts and operations. In fact, disjointness must in any case be enforced by the "pushout construction" for parameter instantiation, implicitly described in what follows for this simple class of uni-parametric parameterized theories.

Suppose now that $B$, $B'$ and $B \cup B'$ have finitary unification algorithms and that both $\mathcal{R}[X] = (\mathcal{R}, X)$ and $\mathcal{G}$ protect, respectively, constructor theories,[14] say $\mathcal{R}_{(\Omega,\Delta)}[X] = (\Omega \cup \Delta, B_{(\Omega,\Delta)}, R_{(\Omega,\Delta)})$ and $\mathcal{G}_{(\Omega',\Delta')} = (\Omega' \cup \Delta', B_{(\Omega',\Delta')}, R_{(\Omega',\Delta')})$. Then $\mathcal{R}[\mathcal{G}, X \mapsto s]$ will protect $\mathcal{R}_{(\Omega,\Delta)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$. Suppose, further, that $B_{(\Omega,\Delta)}$, $B_{(\Omega',\Delta')}$, and $B_{(\Omega,\Delta)} \cup B_{(\Omega',\Delta')}$ have decidable equality.

The general kind of satisfiability-preserving result we are seeking follows the following pattern: (i) assuming that $\mathcal{G}_{(\Omega',\Delta')}$ is the decomposition of an OS-compact theory, then (ii) under some assumptions about the cardinality of the sort $s$, prove the OS-compactness of $\mathcal{R}_{(\Omega,\Delta)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$. By Corollary 2 this then proves that satisfiability of QF formulas in the initial model of the instantiation $\mathcal{R}[\mathcal{G}, X \mapsto s]$ is decidable. Let us see some examples.

*Example 17.* (Lists). This parameterized module $\mathcal{L}[X]$ has parameter sort $X$ and additional sorts *List* and *NeList* (non-empty lists), with subsorts $NeList < List$, constructors $nil :\rightarrow List$ and $\_;\_ : X\ List \rightarrow NeList$, and defined functions $head : NeList \rightarrow X$, and $tail : NeList \rightarrow List$, with defining rules: $head(x; l) \rightarrow x$, and $tail(x; l) \rightarrow l$, where $x$ has sort $X$ and $l$, sort *List*. Subsorts cut through the usual nonsense about expressions like $head(nil)$. Indeed, they solve in an elegant and fully general way the "constructor-selector problem" for data types [75]. This module is FVP with variant complexity 4, is sufficiently complete, and protects its constructor decomposition $\mathcal{L}_\Omega[X]$.

**Theorem 9.** *For $\mathcal{L}[X]$ the above parameterized list module, protecting the obvious constructor decomposition $\mathcal{L}_\Omega[X]$, $\mathcal{G} = (\Sigma' \cup \Pi', B', R')$ an FVP decomposition of a finitary OS-FO equational theory $((\Sigma', \Pi'), \Gamma')$, where $\mathcal{G}$ protects a constructor decomposition $\mathcal{G}_{(\Omega',\Delta')} = (\Omega' \cup \Delta', B_{(\Omega',\Delta')}, R_{(\Omega',\Delta')})$ of an equational OS-FO-compact theory $((\Omega', \Delta'), \Gamma)$, and $s$ an infinite sort of $\mathcal{G}$ in $\Omega'$, if: (i) $\mathcal{L}[X]$ and $\mathcal{G}$ are both p-terminating for a modular termination property p or $\mathcal{L}[\mathcal{G}, X \mapsto s]$ is terminating, (ii) $B'$ has a finitary unification algorithm extensible with free function symbols; and (iii) $B_{(\Omega',\Delta')}$-equality is decidable, then $\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ is the decomposition of an OS-compact theory and therefore satisfiability of QF formulas in the initial model of the instantiation $\mathcal{L}[\mathcal{G}, X \mapsto s]$ is decidable.*

[14] For more details about sufficient completeness of parameterized OS theories and methods for checking it see [70].

*Proof.* We have to show that any axiom-consistent and normalized[15] conjunction of $\bigwedge D$ of $(\Omega \cup \Omega', \Delta')$-disequalities[16] such that all its variables have infinite sorts[17] is satisfiable in the initial model $C_{\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$.

Since it is easy to prove that $\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ protects its parameter $\mathcal{G}_{(\Omega',\Delta')}$, we have $C_{\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}|_{\Omega'\cup\Delta'} = C_{\mathcal{G}_{(\Omega',\Delta')}}$. Therefore, using the OS-compactness of $\mathcal{G}_{(\Omega',\Delta')}$, we will be done if we can exhibit a normalized and axiom-consistent conjunction $\bigwedge D'$ of $(\Omega', \Delta')$-disequalities, with infinite sort variables, whose satisfaction in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ implies that of $\bigwedge D$ in $C_{\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$.

Let $\overline{Y}$ be a sequence of the distinct variables of sort either *List* or *NeList* appearing in $\bigwedge D$, and $\overline{y}$ a corresponding sequence of fresh new variables of sort $s$. Let $\{\overline{Y} \mapsto \overline{y}; nil\}$ denote the substitution mapping each $Y$ in $\overline{Y}$ to the term $y; nil$, where $y$ in $\overline{y}$ is the fresh variable associated to $Y$. It is easy to check that the conjunction $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ is also normalized and axiom-consistent. Furthermore, since it is a substitution instance of $\bigwedge D$, we will be done if we can show that $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ is satisfiable in the initial algebra $C_{\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$. We will be able to show this if we can build, disequation by disequation, a normalized and axiom-consistent conjunction $\bigwedge D'$ of $(\Omega', \Delta')$-disequalities, with infinite sort variables, whose satisfaction in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ implies that of $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ in $C_{\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$. We build $\bigwedge D'$ as follows: any $(\Omega', \Delta')$-disequality is left untouched. Note that (up to symmetry of $\neq$) the remaining disequalities must be of one of the following three forms: (i) $nil \neq u_1; \ldots; u_n; nil$, with the $u_i$ of sort $s$ or less and $n \geqslant 1$; (ii) $u_1; \ldots; u_n; nil \neq v_1; \ldots; v_n; nil$, with each $u_i$ and $v_i$ of sort $s$ or less and $n \geqslant 1$; and (iii) $u_1; \ldots; u_n; nil \neq v_1; \ldots; v_m; nil$, with each $u_i$ and $v_j$ of sort $s$ or less and $n > m \geqslant 1$. Since it is easy to show that disequalities of types (i) and (iii) are *valid* in the initial algebra of $\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$, we can ignore them when building $\bigwedge D'$. But since each disequality of type (ii) is $B_{(\Omega',\Delta')}$-consistent, this means that there must be a $q$, $1 \leqslant q \leqslant n$, such that $u_q \neq_{B_{(\Omega',\Delta')}} v_q$. We then replace that disequality by the disequality $u_q \neq v_q$ in $\bigwedge D'$. By construction $\bigwedge D'$ has variables only of infinite sorts and is both normalized and axiom-consistent. Furthermore, any satisfying assignment for $\bigwedge D'$ in the initial algebra of $\mathcal{G}_{(\Omega',\Delta')}$ extends to a satisfying assignment for $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ in the initial algebra of $\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$, as desired. □

We can consider, for example, the instantiation $\mathcal{L}[\mathcal{Z}_+, X \mapsto Int]$ of the list data type, yielding lists of integers. Since $\mathcal{Z}_+$ satisfies the requirements in The-

[15] Here, and in what follows, by "axiom-consistent and normalized" formula on a given signature I will mean for the axioms and rules of the decomposition having that signature, which in this case is $\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$. So in this case I mean: "$B_{(\Omega',\Delta')}$-consistent and $R_{(\Omega',\Delta')}, B_{(\Omega',\Delta')}$-normalized." Note also the slight abuse of language, since in $(\Omega \cup \Omega' \cup \Delta')$ the signature $\Omega$ has been renamed to $\Omega[X \mapsto s]$, so this notation really abbreviates: $(\Omega[X \mapsto s] \cup \Omega' \cup \Delta')$.

[16] Here, and in what follows, the expression "a conjunction $\bigwedge D$ of $(\Omega \cup \Omega', \Delta')$-disequalities" is shorthand for: "a conjunction $\bigwedge D$ which is the functional version $\bigwedge D = \bigwedge \widetilde{D}_0$ of a conjunction $\bigwedge D_0$ of negated $(\Omega \cup \Omega', \Delta')$-atoms.

[17] Here, and in what follows, the decomposition in which the variables have infinite sorts will be clear from the context. In this case it is of course $\mathcal{L}_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$.

orem 9, QF satisfiability in $C_{\mathcal{L}[\mathcal{Z}_+, X \mapsto Int]}$ is decidable. We can, for example, prove that (for this instantiation and actually for any other[18] satisfying the theorem's conditions) the equality $head(l'); tail(l') = l'$, where $l'$ has sort *NeList*, is an inductive theorem in $C_{\mathcal{L}[\mathcal{Z}_+, X \mapsto Int]}$. This is equivalent to checking that $head(l'); tail(l') \neq l'$ is unsatisfiable. The variants are: $head(l'); tail(l')) \neq l'$, and $i; l \neq i; l$, which is the only constructor variant and, since $\varnothing$-inconsistent, unsatisfiable.

*Example 18.* (Compact Lists). Compact lists are lists with no contiguous repeated elements. They are used for greater efficiency in various constrained logic programming applications [35,34]. Their specification as a parameterized data type $\mathcal{L}^c[X]$ is exactly like that for lists, except for two small changes: (i) we keep the *head* defined function and its rule, but drop the *tail* function and its rule; and (ii) in the protected constructor subspecification $\mathcal{L}^c_\Omega[X]$ we add the following rule between constructors terms: $x; (x; l) \to x; l$, where $x$ has sort $X$ and $l$ has sort *List*. This decomposition of parameterized compact lists is FVP with variant complexity 4, is sufficiently complete, and protects its constructor decomposition $\mathcal{L}^c_\Omega[X]$.

As for lists, we have the following parametric decidability-preserving result:

**Theorem 10.** *For $\mathcal{L}^c[X]$ the above parameterized compact list module, protecting the obvious constructor decomposition $\mathcal{L}^c_\Omega[X]$, $\mathcal{G} = (\Sigma' \cup \Pi', B', R')$ an FVP decomposition of a finitary OS-FO equational theory $((\Sigma', \Pi'), \Gamma')$, where $\mathcal{G}$ protects a constructor decomposition $\mathcal{G}_{(\Omega', \Delta')} = (\Omega' \cup \Delta', B_{(\Omega', \Delta')}, R_{(\Omega', \Delta')})$ of an equational OS-FO-compact theory $((\Omega', \cup \Delta'), \Gamma)$, and $s$ an infinite sort of $\mathcal{G}$ in $\Omega'$, if: (i) $\mathcal{L}^c[X]$ and $\mathcal{G}$ are both p-terminating for a modular termination property $p$ or $\mathcal{L}^c[\mathcal{G}, X \mapsto s]$ is terminating, (ii) $B'$ has a finitary unification algorithm extensible with free function symbols; and (iii) $B_{(\Omega', \Delta')}$-equality is decidable, then $\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega', \Delta')}, X \mapsto s]$ is the decomposition of an OS-compact theory and therefore satisfiability of QF formulas in the initial model of the instantiation $\mathcal{L}^c[\mathcal{G}, X \mapsto s]$ is decidable.*

*Proof.* We have to show that any normalized and axiom-consistent conjunction $\bigwedge D$ of $(\Omega \cup \Omega', \Delta')$-disequalities whose variables have infinite sorts is satisfiable in the initial algebra $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega', \Delta')}, X \mapsto s]}$. Since $\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega', \Delta')}, X \mapsto s]$ protects its parameter $\mathcal{G}_{(\Omega', \Delta')}$, we have $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega', \Delta')}, X \mapsto s]}|_{\Omega' \cup \Delta'} = C_{\mathcal{G}_{(\Omega', \Delta')}}$. Therefore, using the OS-compactness of $\mathcal{G}_{(\Omega', \Delta')}$, we will be done if we can exhibit a normalized and axiom-consistent conjunction $\bigwedge D'$ of $(\Omega', \Delta')$-disequalities, with infinite sort variables, whose satisfaction in $C_{\mathcal{G}_{(\Omega', \Delta')}}$ implies that of $\bigwedge D$ in $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega', \Delta')}, X \mapsto s]}$.

[18] In what follows I will discuss several formulas that are "parametric theorems," valid in any correct instantiation of various parameterized data types. However, in this paper I will always do so in the context of a *concrete instantiation*. The details of the "parametric proof method" where we reason directly and generically in the parameterized theory $\mathcal{L}[X]$ itself, in the style of [70], will be developed elsewhere.

Let $\overline{Y}$ be a sequence of the distinct variables of sort either *List* or *NeList* appearing in $\bigwedge D$, and $\overline{y}$ a corresponding sequence of fresh new variables of sort $s$. Let $\{\overline{Y} \mapsto \overline{y}; nil\}$ denote the substitution mapping each $Y$ in $\overline{Y}$ to the term $y; nil$, where $y$ in $\overline{y}$ is the fresh variable associated to $Y$. It is easy to check that the conjunction $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ is also normalized and axiom-consistent. Furthermore, since it is a substitution instance of $\bigwedge D$, we will be done if we can show that $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ is satisfiable in the initial model $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$. We will be able to show this if we can build, disequation by disequation, a normalized and axiom-consistent conjunction $\bigwedge D'$ of $(\Omega',\Delta')$-disequalities, with infinite sort variables, whose satisfaction in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ implies that of $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ in $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$. We build $\bigwedge D'$ as follows. Any $(\Omega',\Delta')$-disequality is left untouched. Note that (up to symmetry of $\neq$) the remaining disequalities must be of one of the following three forms: (i) $nil \neq u_1; \ldots; u_n; nil$, with the $u_i$ of sort $s$ or less and $n \geqslant 1$, which is a valid disequality in $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$ and can therefore be ignored; (ii) $u_1; \ldots; u_n; nil \neq v_1; \ldots; v_n; nil$, with each $u_i$ and $v_i$ of sort $s$ or less and $n \geqslant 1$, where by irreducibility we must have $u_i \neq_{B_{(\Omega',\Delta')}} u_{i+1}$, $1 \leqslant i < n$ and $v_i \neq_{B_{(\Omega',\Delta')}} v_{i+1}$, $1 \leqslant i < n$, and by axiom-consistency there must be a $u_q \neq_{B_{(\Omega',\Delta')}} v_q$ for some $1 \leqslant q \leqslant n$; we can then replace $u_1; \ldots; u_n; nil \neq v_1; \ldots; v_n; nil$ by the conjunction $u_q \neq v_q \wedge \bigwedge_{1\leqslant i<n} u_i \neq u_{i+1} \wedge \bigwedge_{1\leqslant i<n} v_i \neq v_{i+1}$; and (iii) $u_1; \ldots; u_n; nil \neq v_1; \ldots; v_m; nil$, with each $u_i$ and $v_j$ of sort $s$ or less and $n \neq m$, where by irreducibility we must have $u_i \neq_{B_{(\Omega',\Delta')}} u_{i+1}$, $1 \leqslant i < n$ and $v_j \neq_{B_{(\Omega',\Delta')}} v_{j+1}$, $1 \leqslant i < m$, and then we can replace $u_1; \ldots; u_n; nil \neq v_1; \ldots; v_m; nil$ by the conjunction $\bigwedge_{1\leqslant i<n} u_i \neq u_{i+1} \wedge \bigwedge_{1\leqslant j<m} v_j \neq v_{j+1}$. In this way we obtain a conjunction $\bigwedge D'$ which, by construction, has variables only of infinite sorts and is both normalized and axiom-consistent. Furthermore, any satisfying assignment for $\bigwedge D'$ in the initial model $C_{\mathcal{G}_{(\Omega',\Delta')}}$ extends to a satisfying assignment for $\bigwedge D\{\overline{Y} \mapsto \overline{y}; nil\}$ in the initial model $C_{\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$, as desired. □

Since for $\mathcal{G}$ either $\mathcal{Z}_{s,p}$ or $\mathcal{Z}_+$ the conditions in the above theorem are met when the parameter sort $X$ is instantiated to the *Int* sort, validity of QF formulas in the initial algebra of $\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto Int]$ for compact list of integers in either of these two instantiations is decidable. For example, the following simple theorem holds for any instantiation satisfying the above requirements, and does so, in particular, for $\mathcal{L}^c_\Omega[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto Int]$ for compact list of offset integers: $x; l = l \Rightarrow head(l) = x$. To show it we just need to prove that $x; l = l \wedge head(l) \neq x$ is unsatisfiable. Variant unification gives us the single unifier $\{l \mapsto x; l'\}$ for the equation $x; l = l$, yielding the disequality $head(x; l') \neq x$, which normalizes to the inconsistent disequality $x \neq x$.

*Example 19.* (Multisets). Let $\mathcal{M}[X]$ be the following FVP decomposition. There is first a parameterized constructor FVP decomposition $\mathcal{M}_{(\Omega,\Pi)}[X]$ whose signature $\Omega$ of constructors has a parameter sort $X$, a sort *MSet*, representing multisets, a sort *NeMSet*, representing non-empty multisets, and subsort inclusions $X < NeMSet < MSet$. The *constructors* are: (i) a constant $\varnothing :\rightarrow MSet$, and a multiset union operator $\_,\_ : NeMSet\ NeMSet \rightarrow NeMSet$, which is

given $AC$ axioms. The signature $\Pi$ of constructor *predicates* is represented in $\mathcal{M}_{(\Omega,\Pi)}[X]$ by a sort *Pred*, a constant $tt :\rightarrow Pred$, the membership predicate $\_ \in \_ : X\ MSet \rightarrow Pred$, and a predicate for multisets with duplicated elements, $dupl : MSet \rightarrow Pred$.

The decomposition $\mathcal{M}_{(\Omega,\Pi)}[X]$ has the $AC$ axioms for union as only axioms, and its rules $R_{(\Omega,\Pi)}$ are: (i) those defining the $\in$ predicate, namely, for $x$ a variable of sort $X$ and $M, M'$ variables of sort *NeMSet*, the axiom $x \in x$, represented by the rule $x \in x \rightarrow tt$, and the axiom $x \in x, M$ represented by the rule $x \in x, M \rightarrow tt$; and (ii) rules defining the *dupl* predicate, with the axiom $dupl(x, x)$ represented by the rule $dupl(x, x) \rightarrow tt$, and the axiom $dupl(x, x, M)$ represented by the rule $dupl(x, x, M) \rightarrow tt$.

$\mathcal{M}[X]$ extends $\mathcal{M}_{(\Omega,\Pi)}[X]$ in a sufficiently complete and protecting mode by adding a defined function symbol $\_,\_ : MSet\ MSet \rightarrow MSet$ satisfying also the $AC$ axioms, and having the identity rule, $Q, \varnothing \rightarrow Q$, with $Q$ a variable of sort *MSet*. The module $\mathcal{M}[X]$ is FVP with variant complexity 9.

Here is now a parametric, decidable QF satisfiability result for multiset instances $\mathcal{M}[\mathcal{G}, X \mapsto s]$.

**Theorem 11.** *For $\mathcal{M}[X]$ the above parameterized multiset module, protecting the constructor decomposition $\mathcal{M}_{(\Omega,\Pi)}[X]$, $\mathcal{G} = (\Sigma' \cup \Pi', B', R')$ an FVP decomposition of a finitary OS-FO equational theory $((\Sigma', \Pi'), \Gamma')$, where $\mathcal{G}$ protects a constructor decomposition $\mathcal{G}_{(\Omega',\Delta')} = (\Omega' \cup \Delta', B_{(\Omega',\Delta')}, R_{(\Omega',\Delta')})$ of an equational OS-FO-compact theory $((\Omega', \Delta'), \Gamma)$, and $s$ an infinite sort of $\mathcal{G}$ in $\Omega'$, if: (i) $\mathcal{M}[X]$ and $\mathcal{G}$ are both p-terminating for a modular termination property p or $\mathcal{M}[\mathcal{G}, X \mapsto s]$ is terminating, (ii) $B'$ and $B' \cup AC$ have finitary unification algorithms and (iii) $B_{(\Omega',\Delta')} \cup AC$-equality is decidable, then $\mathcal{M}_{\Omega,\Pi}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ is the decomposition of an OS-compact theory and therefore satisfiability of QF formulas in the initial model of the instantiation $\mathcal{M}[\mathcal{G}, X \mapsto s]$ is decidable.*

*Proof.* First of all note that the finite sorts of $\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ are exactly those of $\mathcal{G}_{(\Omega',\Delta')}$. We have to prove that any axiom-consistent normalized conjunction of $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities $\bigwedge D'$ whose variables have all infinite sorts is satisfiable in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$.

Let $\overline{Y}$ be all the variables of sort *NeMSet* or *MSet* in $\bigwedge D'$, and let $\overline{y}$ be a corresponding set of fresh variables of sort $s$. Since $\bigwedge D'\{\overline{Y} \mapsto \overline{y}\}$ is a substitution instance of $\bigwedge D'$ and it is easy to check that it is normalized and axiom-consistent, we will be done if we show that $\bigwedge D'\{\overline{Y} \mapsto \overline{y}\}$ is satisfiable in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$. Since it is also easy to prove that $\mathcal{M}_{\Omega}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ protects its parameter $\mathcal{G}_{(\Omega',\Delta')}$, so that $C_{\mathcal{M}_{\Omega}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}|_{\Omega' \cup \Delta'} = C_{\mathcal{G}_{(\Omega',\Delta')}}$, we will be done, thanks to OS-compactness, if we can build, disequation by disequation, a normalized and axiom-consistent conjunction $\bigwedge D'$ of $(\Omega', \Delta')$-disequalities whose variables have all infinite sorts, and whose satisfaction in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ implies that of $\bigwedge D\{\overline{Y} \mapsto \overline{y}\}$ in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$. We can do so by leaving all $(\Omega', \Delta')$-disequations untouched and replacing any other disequation $u \neq v$ of $\bigwedge D'\{\overline{Y} \mapsto \overline{y}\}$ either by nothing if it is valid in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$,

or by a conjunction $C$ of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities such that $C \Rightarrow u \neq v$ is a *valid* formula in $C_{\mathcal{M}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$. We have two kinds of such disequalities: those between terms of sort *MSet* or less, and those between negated atoms of sort *Pred*.

Up to symmetry of $\neq$, and up to $AC \cup B_{(\Omega',\Delta')}$-equality, normalized and axiom-consistent disequalities of sort *MSet* or less in $\bigwedge D'\{\overline{Y} \mapsto \overline{y}\}$ that are not $(\Omega', \Delta')$-formulas must be of one of the following forms: (i) $u \neq \varnothing$, with $u$ of sort *NeMSet* or less, which is a *valid* disequality in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$, and therefore can be *ignored*; or (ii) $n_1 \cdot u_1, \ldots, n_k \cdot u_k \neq m_1 \cdot v_1, \ldots, m_l \cdot v_l$, where: (1) $n \cdot w$, $n \geqslant 1$, abbreviates the multiset $w, \overset{n}{.\,.}, w$, (2) $\Sigma n_i + \Sigma m_j \geqslant 3$, (3) the $u_i$ and $v_j$ have sort $s$ or less, and (4) if $i \neq i'$, then $u_i \neq_{B_{(\Omega',\Delta')}} u_{i'}$, and if $j \neq j'$, then $v_j \neq_{B_{(\Omega',\Delta')}} v_{j'}$. If $\Sigma n_i \neq \Sigma n_j$, the disequality is valid in $C_{\mathcal{M}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$, and therefore can be ignored. If $\Sigma n_i = \Sigma n_j$ and $k > l$ (the case $k < l$ is similar) there must be a $u_i$ such that $u_i \neq_{B_{(\Omega',\Delta')}} v_j$, $1 \leqslant j \leqslant l$, and we can replace $n_1 \cdot u_1, \ldots, n_k \cdot u_k \neq m_1 \cdot v_1, \ldots, m_l \cdot v_l$ by the conjunction of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{1 \leqslant j \leqslant l} u_i \neq v_j$. If $\Sigma n_i = \Sigma n_j$ and $k = l$, there must be a $u_i$ such that $u_i \neq_{B_{(\Omega',\Delta')}} v_j$, $1 \leqslant j \leqslant l$, since otherwise $n_1 \cdot u_1, \ldots, n_k \cdot u_k \neq m_1 \cdot v_1, \ldots, m_l \cdot v_l$ would be $AC \cup B_{(\Omega',\Delta')}$-inconsistent. Therefore we can replace it by the conjunction of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{1 \leqslant j \leqslant l} u_i \neq v_j$.

Up to symmetry of $\neq$, and up to $AC \cup B_{(\Omega',\Delta')}$-equality, normalized and axiom-consistent disequalities of sort *Pred* in $\bigwedge D'\{\overline{Y} \mapsto \overline{y}\}$ that are not $(\Omega', \Delta')$-formulas must be of one of the following forms: (i) $u \in \varnothing \neq tt$, with $u$ of sort $s$ or less, which is valid in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$ and therefore can be ignored; (ii) $u \in n_1 \cdot u_1, \ldots, n_k \cdot u_k \neq tt$, $\Sigma n_i \geqslant 1$, where the $u_1, \ldots, u_k$ are terms of sort $s$ or less $B_{(\Omega',\Delta')}$-different among themselves, and, by normalization and axiom-coherence, we must have $u \neq_{B_{(\Omega',\Delta')}} u_i$, $1 \leqslant i \leqslant k$. We can then replace $u \in n_1 \cdot u_1, \ldots, n_k \cdot u_k \neq tt$ by the conjunction of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{1 \leqslant i \leqslant k} u \neq u_i$. (iii) $dupl(\varnothing) \neq tt$, or $dupl(u) \neq tt$, with $u$ of sort $s$ or less, which are both valid in $C_{\mathcal{M}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$, and therefore can be ignored; or (iv) $dupl(u_1, \ldots, u_k) \neq tt$, where $k \geqslant 2$ and the $u_1, \ldots, u_k$ are terms of sort $s$ or less $B_{(\Omega',\Delta')}$-different among themselves. We can then replace $dupl(u_1, \ldots, u_k) \neq tt$ by the conjunction of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{i \neq j} u_i \neq u_j$. In this way we obtain our desired conjunction $\bigwedge D'$ of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities whose variables have all infinite sorts, and whose satisfaction in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ implies that of $\bigwedge D\{\overline{Y} \mapsto \overline{y}\}$ in $C_{\mathcal{M}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$. □

The requirement that $s$ is an infinite sort is essential for the multiset parameterized module to preserve OS-compactness. Otherwise, $C_{\mathcal{G},s} = \{[u_1], \ldots, [u_n]\}$ for some $n \geqslant 1$, and for $M$ a variable of sort *NeMSet* the normalized and $AC \cup B_{(\Omega',\Delta')}$-consistent conjunction of disequalities

$$u_1 \in M \neq tt \ \wedge \ \ldots \ \wedge \ u_n \in M \neq tt$$

is unsatisfiable.

Note that the conditions in Theorem 11 apply for $\mathcal{G}$ the FVP decompositions $\mathcal{Z}_{s,p}$ of offset integers and $\mathcal{Z}_{+}$ of integers with addition when the parameter $X$ is mapped to the sort *Int*. Perhaps more interestingly, thanks to Theorem 9, the conditions in Theorem 11 also apply for $\mathcal{G} = \mathcal{L}[\mathcal{G}', X \mapsto s]$, where the parameter $X$ for the List module is mapped to an infinite sort $s$ in any FVP decomposition $\mathcal{G}'$ protecting an OS-compact constructor decomposition. For example, when $\mathcal{G}'$ is either $\mathcal{Z}_{s,p}$ or $\mathcal{Z}_{+}$ and $s = \mathit{Int}$. In this way we get decidable satisfiability for *multisets of lists of integers*, or, more generally, multisets of lists of *anything* FVP with a compact constructor decomposition and with an infinite sort $s$.

Let us see an example of a valid theorem in $\mathcal{M}[\mathcal{Z}_{s,p}, X \mapsto \mathit{Int}]$ (in fact the theorem in question does not involve the language of offset integers and is a *generic* theorem of the $\mathcal{M}[X]$ parameterized module). To see that for $x, y$ of sort *Int* and $M$ and $M'$ of sort *MSet*, the formula

$$(x \in M = tt \ \wedge \ M = y, M' \ \wedge \ x \neq y) \Rightarrow x \in M' = tt$$

is valid in the initial model of $\mathcal{M}[\mathcal{Z}_{s,p}, X \mapsto \mathit{Int}]$, we just need to show that its negation

$$x \in M = tt \ \wedge \ M = y, M' \ \wedge \ x \neq y \ \wedge x \in M' \neq tt$$

is unsatisfiable in such an initial model. The variant unification of $x \in M = tt \ \wedge \ M = y, M'$ yields three unifiers: $\{(M \mapsto x, y), (M' \mapsto x)\}$, $\{(M \mapsto x, y, M''), (M' \mapsto x, M'')\}$, and $\{(M \mapsto x, M'), (y \mapsto x)\}$, which yield the respective three conjunctions of disequalities: $x \neq y \ \wedge \ x \in x, y \neq tt$, and $x \neq y \ \wedge \ x \in x, M'' \neq tt$, and $x \neq x \ \wedge \ x \in M'' \neq tt$. The last one is *AC*-inconsistent; the first two become so by simplification with the rules for $\in$.

*Example 20.* (Sets). $\mathcal{S}[X]$ is a parameterized module whose signature and rules are those for multisets, except that: (i) we rename the sorts *NeMSet* and *MSet* to, respectively, *NeSet* and *Set*; (ii) we drop the $\in$ and *dupl* predicates and their rules and add instead the constructor predicate $\_ \subseteq \_ : \mathit{Set}\ \mathit{Set} \rightarrow \mathit{Pred}$; (iii) for $S, S'$ variables of sort *NeSet* add the "idempotency" rules $S, S \rightarrow S$ and $S, S, S' \rightarrow S, S'$; and (iv) for $U, V$ variables of sort *Set* define $\subseteq$ by the rules: $\varnothing \subseteq U \rightarrow tt$, $U \subseteq U \rightarrow tt$, and $U \subseteq U, V \rightarrow tt$. This parameterized decomposition of sets is FVP with variant complexity 11 and sufficiently complete, and protects the constructor decomposition $\mathcal{S}_{\Omega,\Pi}[X]$.

The predicates $\in$ and $\subset$ need not be explicitly defined, since they can be expressed by the definitional equivalences $x \in U = tt \Leftrightarrow x, U = U$, with $x$ of sort $s$, and $U \subset V = tt \Leftrightarrow U \subseteq V = tt \ \wedge \ U \neq V$.

As for multisets, but with a broader scope of instances, we have the following, general decidable QF satisfiability result for instances $\mathcal{S}[\mathcal{G}, X \mapsto s]$ of the set parameterized module. It uses the auxiliary notion of an *infinity-closed* decomposition $\mathcal{G}$, defined as a theory where, if a term $t$ has at least one variable having an infinite sort, then the least sort of $t$ is itself infinite. For example, offset integers have the *Zero* finite sort, but are infinity-closed.

**Theorem 12.** *For $\mathcal{S}[X]$ the above parameterized set module, protecting the constructor decomposition $\mathcal{S}_{(\Omega,\Pi)}[X]$, $\mathcal{G} = (\Sigma' \cup \Pi', B', R')$ an infinity-closed FVP decomposition of a finitary OS-FO equational theory $((\Sigma', \Pi'), \Gamma')$, where $\mathcal{G}$ protects an FVP constructor decomposition $\mathcal{G}_{(\Omega',\Delta')} = (\Omega' \cup \Delta', B_{(\Omega',\Delta')}, R_{(\Omega',\Delta')})$ of an equational OS-FO-compact theory $((\Omega', \Delta'), \Gamma)$, and $s$ is a sort of $\mathcal{G}$ in $\Omega'$, if: (i) either $\mathcal{S}[X]$ and $\mathcal{G}$ are both p-terminating for a modular termination property p or $\mathcal{S}[\mathcal{G}, X \mapsto s]$ is terminating; (ii) $B_{(\Omega',\Delta')}$, $B'$ and $B' \cup AC$ have finitary unification algorithms, and (iii) $B_{(\Omega',\Delta')} \cup AC$-equality is decidable, then $\mathcal{S}_{\Omega,\Pi}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ is the decomposition of an OS-compact theory and therefore satisfiability of QF formulas in the initial model of the instantiation $\mathcal{S}[\mathcal{G}, X \mapsto s]$ is decidable.*

*Proof.* We must prove the result for two cases: when $s$ is finite, and when it is infinite. In the first case, note that the infinite sorts of $\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ are exactly those of $\mathcal{G}_{(\Omega',\Delta')}$. $\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ is then OS-compact because: (i) the $\mathcal{G}_{(\Omega',\Delta')}$ part is protected, (ii) $\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ has a finitary unification algorithm and decidable equality, and (iii) since $\mathcal{G}$ is infinity-closed and all the infinite sorts are in $\mathcal{G}$, any normalized and axiom-consistent conjunction of $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities whose variables have all infinite sorts must necessarily decompose into three conjunctions: $\bigwedge D \wedge \bigwedge D' \wedge \bigwedge D''$, where: (i) $\bigwedge D$ is an $(\Omega', \Delta')$-formula and therefore satisfiable, $\bigwedge D'$ is a conjunction of disequations that, up to symmetry, have the form $u \neq v$, with $u$ an $\Omega'$-term having a non-empty set of variables of infinite sorts, and $v$ an normalized *ground term* of sort either *NeSet* or *Set*, which is a *valid* conjunction, because the variants of $u$ can never have sort *NeSet* or *Set*, and (iii) $\bigwedge D''$ a conjunction of ground and axiom-consistent normalized $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities and therefore *valid* in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$.

Assume now that $s$ is infinite. Then the finite sorts of $\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ are exactly those of $\mathcal{G}_{(\Omega',\Delta')}$. We have to prove that any axiom-consistent normalized conjunction of $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities $\bigwedge D$ whose variables have all infinite sorts is satisfiable in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$.

Let $\overline{Y}$ be all the variables of sort *NeSet* or *Set* in $\bigwedge D$, and let $\overline{y}$ be a corresponding set of fresh variables of sort $s$. Since $\bigwedge D\{\overline{Y} \mapsto \overline{y}\}$ is a substitution instance of $\bigwedge D$, and it is easy to check that is normalized, axiom-consistent, we will be done if we show that $\bigwedge D\{\overline{Y} \mapsto \overline{y}\}$ is satisfiable in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$. Since it is easy to prove that $\mathcal{S}_{\Omega}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ protects its parameter $\mathcal{G}_{(\Omega',\Delta')}$, so that we have $C_{\mathcal{S}_{\Omega}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}|_{\Omega' \cup \Delta'} = C_{\mathcal{G}_{(\Omega',\Delta')}}$, we will be done, thanks to OS-compactness, if we can build, disequation by disequation, a normalized and axiom-consistent conjunction $\bigwedge D'$ of $(\Omega', \Delta')$-disequalities whose variables have all infinite sorts, and whose satisfaction in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ implies that of $\bigwedge D\{\overline{Y} \mapsto \overline{y}\}$ in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$. We can do so by replacing each disequation $u \neq v$ in $D\{\overline{Y} \mapsto \overline{y}\}$ which is not a $(\Omega', \Delta')$-disequality either by *nothing* if it is valid in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$, or by a conjunction $C$ of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities in $\mathcal{G}_{(\Omega',\Delta')}$ such that $C \Rightarrow u \neq v$ is a *valid* formula in

$C_{\mathcal{S}_\Omega[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]}$. We have two kinds of such disequalities: those between terms of sort *Set* or less, and those between negated atoms of sort *Pred*.

Up to symmetry of $\neq$, and up to $AC \cup B_{(\Omega',\Delta')}$-equality, normalized and axiom-consistent disequalities in $\bigwedge D\{\overline{Y} \mapsto \overline{y}\}$ between terms of sort *Set* or less, and not $(\Omega', \Delta')$-formulas, must be such that at least one of the terms has sort no lower than *NeSet* and must have one of the following forms: (i) $u \neq \emptyset$, with $u$ of sort *NeSet* or less, which is a *valid* disequality in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]|_{\Omega'\cup\Delta'}}$ and therefore can be *ignored*; or (ii) $\overline{u}_k, \overline{v}_n \neq \overline{u}_k, \overline{w}_m$, where, by convention, $\overline{p}_n$, $n \geqslant 0$, abbreviates the term $\overline{p}_n = p_1, \ldots, p_n$, which vanishes (is not there at all) for $n = 0$, $\overline{u}_k$, $k \geqslant 0$, represents the "maximally shared part" between the two sides, and: (1) all individual terms in $\overline{u}_k, \overline{v}_n, \overline{w}_m$, have all sort $s$ or less and are all $AC \cup B_{(\Omega',\Delta')}$-different from each other (i.e., $\overline{u}_k$, $\overline{v}_n$ and $\overline{w}_m$ represent *mutually disjoint* sets in normalized form), (2) $n + m \geqslant 1$, and if if $k = 0$ we must have $n + m \geqslant 3$. Wen $k = 0$ we replace the formula by the conjunction of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{i,j} v_i \neq w_j$; and when $k > 0$ by the conjunctions of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{i,j} v_i \neq w_j \wedge \bigwedge_{l,i} u_l \neq v_i \wedge \bigwedge_{l,j} u_l \neq w_j$ (two of the conjuncts will be missing if $n + m = 1$).

Up to $AC \cup B_{(\Omega',\Delta')}$-equality, normalized and axiom-consistent disequalities of sort *Pred* in $\bigwedge D'\{\overline{Y} \mapsto \overline{y}\}$ and not $(\Omega', \Delta')$-formulas must be of one of the following forms: (i) $\overline{u}_k \subseteq \emptyset \neq tt$, $k \geqslant 1$, where the $\overline{u}_i$ have sort $s$ or less, which is a *valid* disequality in $C_{\mathcal{S}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X\mapsto s]|_{\Omega'\cup\Delta'}}$, and therefore can be ignored; or (ii) $\overline{u}_n \subseteq \overline{v}_m \neq tt$, $n, m \geqslant 1$, such that $(\exists i)\ \bigwedge_{1\leqslant j\leqslant m} u_i \neq_{AC\cup B_{(\Omega',\Delta')}} v_j$, since otherwise the negation would give us $(\forall i)\ \bigvee_{1\leqslant j\leqslant m} u_i =_{AC\cup B_{(\Omega',\Delta')}} v_j$, violating the irreducibility assumption. Therefore, we can replace this disequality by the conjunction of normalized and axiom-consistent $(\Omega', \Delta')$-disequalities $\bigwedge_{1\leqslant j\leqslant m} u_i \neq v_j$. □

This theorem gives us decidable QF satisfiability, and therefore decidable QF validity, in the initial model of any instance satisfying the requirements in the theorem. For example, for $\mathcal{S}[\mathcal{Z}_{s,p}, X \mapsto Int]$ sets of offset integers, the formula (again, a generic one valid also for all instantiations meeting the requirements in the theorem), $(x \in y, S = tt \wedge x \neq y) \Rightarrow x \in S = tt$, where $x, y$ have sort *Int*, and $S$ sort *NeSet*, is valid in $C_{\mathcal{S}[\mathcal{Z}_{s,p},X\mapsto Int]}$. This is so because, desugared, it is just $(x, y, S = y, S \wedge x \neq y) \Rightarrow x, S = S$, and its negation, $x, y, S = y, S \wedge x \neq y \wedge x, S \neq S$ is such that the equation $x, y, S = y, S$ has three variant unifiers: $\{S \mapsto x, S'\}$, $\{x \mapsto y\}$, and $\{S \mapsto x\}$, yielding the three conjunctions of disequalities: $x \neq y \wedge x, x, S' \neq s, S'$, and $y \neq y \wedge y, S \neq S$, and $x \neq y \wedge x, x \neq x$. The second is $AC$-inconsistent, and so are the other two when normalized.

*Example 21.* (Hereditarily Finite (HF) Sets). HF sets are a model of set theory without the axiom of infinity. All effective constructions of finitary mathematics —including in particular all effective arithmetic constructions— can be represented within it (see [27],Ch. I). I specify below a data type of HF sets with set union $\cup$ and a set inclusion predicate $\subseteq$ (the predicates $\subset$ and $\in$ are obtained as definitional extensions). As is well-known, all HF sets can be built "ex nihilo"

out of the empty set $\varnothing$. However, it is very convenient to also allow "urelements," like $a, b, c, 7, 2/9, \sqrt{2}, \pi$, and so on, as set elements. This can be achieved by making HF sets parametric on a parameter sort $X$ for such "urelements." That is, HF sets are an FVP parametererized data type $\mathcal{H}[X]$ protecting an FVP constructor subtheory $\mathcal{H}_{(\Omega,\Pi)}[X]$ which has the following signature $\Omega$ of constructors: there are five sorts: $X$, *Elt*, *Set*, *Magma*, and *Pred*, and subsort inclusions $X$ *Set* $<$ *Elt* $<$ *Magma*, where *Magma* represents multisets of sets and has an $AC$ multiset union constructor $\_,\_ : \mathit{Magma}\ \mathit{Magma} \rightarrow \mathit{Magma}$. There is also the empty set constructor constant $\varnothing :\rightarrow \mathit{Set}$, and a constructor $\{\_\} : \mathit{Magma} \rightarrow \mathit{Set}$ that builds a set out of a magma. The signature $\Pi$ of constructor predicates has the usual constructor constant $tt :\rightarrow \mathit{Pred}$, plus the constructor set inclusion predicate $\_ \subseteq \_ : \mathit{Set}\ \mathit{Set} \rightarrow \mathit{Pred}$. Using $M, M'$ as variables of sort *Magma* and $U, V$ as variables of sort *Set*, the rules $R_{(\Omega,\Pi)}$ rewriting constructor terms and constructor predicates are: (i) the"magma idempotency" rules, $M, M \rightarrow M$ and $M, M, M' \rightarrow M, M'$; and (ii) the rules defining the $\subseteq$ predicate, $\varnothing \subseteq U \rightarrow tt$, $\{M\} \subseteq \{M\} \rightarrow tt$, and $\{M\} \subseteq \{M, M'\} \rightarrow tt$.

This constructor decomposition $\mathcal{H}_{(\Omega,\Pi)}[X]$ is extended in a sufficiently complete and *protecting* way by the specification of the union operator $\_ \cup \_ :$ *Set Set* $\rightarrow$ *Set* as a function defined by means of the following rules: $U \cup \varnothing \rightarrow U$, $\varnothing \cup U \rightarrow U$, and $\{M\} \cup \{M'\} \rightarrow \{M, M'\}$. The variant complexity of this decomposition of HF sets is 17.

The predicates $\in$ and $\subset$ need not be explicitly defined, since they can be expressed by the definitional equivalences $x \in V = tt \Leftrightarrow \{x\} \cup V = V$, with $x$ of sort *Elt*, and $U \subset V = tt \Leftrightarrow (U \subseteq V = tt \wedge U \neq V)$.

The expected parameterized preservation of OS-compactness for HF sets can be stated as follows:

**Theorem 13.** *For $\mathcal{H}[X]$ the above parameterized HF set module, protecting the constructor decomposition $\mathcal{H}_{(\Omega,\Pi)}[X]$, $\mathcal{G} = (\Sigma' \cup \Pi', B', R')$ an infinity-closed FVP decomposition of a finitary OS-FO equational theory $((\Sigma', \Pi'), \Gamma')$, where $\mathcal{G}$ protects a constructor decomposition $\mathcal{G}_{(\Omega',\Delta')} = (\Omega' \cup \Delta', B_{(\Omega',\Delta')}, R_{(\Omega',\Delta')})$ of an equational OS-FO-compact theory $((\Omega', \Delta'), \Gamma)$, and $s$ a sort of $\mathcal{G}$ in $\Omega'$, if: (i) $\mathcal{H}[X]$ and $\mathcal{G}$ are both p-terminating for a modular termination property p or $\mathcal{H}[\mathcal{G}, X \mapsto s]$ is terminating, (ii) $B'$ and $B' \cup AC$ have finitary unification algorithms and (iii) $B_{(\Omega',\Delta')} \cup AC$-equality is decidable, then $\mathcal{H}_{\Omega,\Pi}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ is the decomposition of an OS-compact theory and therefore satisfiability of QF formulas in the initial model of the instantiation $\mathcal{H}[\mathcal{G}, X \mapsto s]$ is decidable.*

*Proof.* First of all note that the finite sorts of $\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ are exactly those of $\mathcal{G}_{(\Omega',\Delta')}$. Note also that $\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$ protects its parameter, so that we have $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}|_{(\Omega',\Delta')} = C_{\mathcal{G}_{(\Omega',\Delta')}}$. We have to prove that any axiom-consistent and normalized conjunction of $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities $\bigwedge D$ whose variables have all infinite sorts is satisfiable in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]}$.

Let $\overline{Y}$ be all the variables of sort *Magma*, *Elt*, or *Set* in $\bigwedge D$, and let $\overline{y}$ be a corresponding set of fresh variables of sort *Set*. Assuming that lower case

letters correspond to upper case ones, let $\bigwedge D\{\overline{Y} \mapsto \{\overline{y}\}\}$ denote the substitution instance of $\bigwedge D$ where $Y \mapsto \{y\}$. Note that all the sorts of the variables in $\bigwedge D\{\overline{Y} \mapsto \{\overline{y}\}\}$ are infinite. I claim that $\bigwedge D\{\overline{Y} \mapsto \{\overline{y}\}\}$ is also normalized and axiom-consistent. This follows immediately from the following lemma, whose somewhat lengthy proof is exiled to Appendix C.

**Lemma 1.** *For the equational version $t \neq t'$ of any normalized and axiom-consistent negated $(\Omega \cup \Omega', \Pi \cup \Delta')$-FO-atom, its substitution instance $t\{\overline{Y} \mapsto \{\overline{y}\}\} \neq t'\{\overline{Y} \mapsto \{\overline{y}\}\}$ is also normalized and axiom-consistent.*

By the above lemma we will be done if we show that any normalized and axiom-consistent conjunction of $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities $\bigwedge D'$ that, like $\bigwedge D\{\overline{Y} \mapsto \{\overline{y}\}\}$, has no variables of sorts *Elt* or *Magma*, and where any occurrence of a variable $y$ of sort *Set* must appear within a singleton set subterm $\{y\}$, and where all variables have infinite sorts, is satisfiable in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$. We can do so by induction on $\mu(\bigwedge D') = max(\{|u| + |v| \mid (u \neq v) \in D' \wedge (u \notin T_{\Omega' \cup \Delta'}(X) \vee v \notin T_{\Omega' \cup \Delta'}(X))\})$.

For $\mu(\bigwedge D') = 2$, the only normalized and axiom-consistent disequalities possible are, up to symmetry and $AC \cup B_{(\Omega',\Delta')}$-equality, either: (i) those in $\mathcal{G}_{(\Omega',\Delta')}$, or (ii) $z \neq \varnothing$ with $z$ a variable or constant of sort $s$ or less, which is a valid disequality in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$. Since case (ii) are valid disequalities, and disequalities of case (i) yield a conjunction of disequalities with variables of infinite sorts satisfiable in $C_{\mathcal{G}_{(\Omega',\Delta')}}$ by compactness, and therefore in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$, the base case is proved.

To prove the induction step, assume that the result holds for such normalized and axiom-consistent conjunctions of disequalities whose measure $\mu$ yields any value less or equal to $n$, and let $\mu(\bigwedge D') = n + 1$. We will be done if we can build another conjunction of disequalities $\bigwedge D''$ satisfying the same requirements as $\bigwedge D'$ and such that $\mu(\bigwedge D'') \leqslant n$. We do so disequation by disequation. Disequations between terms of sort *Magma* or less not in $\mathcal{G}_{(\Omega',\Delta')}$ must, ignoring symmetry and up to $AC \cup B_{(\Omega',\Delta')}$-equality, be of one of the following forms: (i) $\varnothing \neq u_1, \ldots, u_n$, $(n \geqslant 1)$, with the $u_i$ of sort either *Set*, or $s$ or less, and if $n = 1$ with $u_1$ of sort $s$ or less; (ii) $\varnothing \neq \{u_1, \ldots, u_n\}$, $(n \geqslant 1)$, with the $u_i$ of sort either *Set*, or $s$ or less; (iii) $u_1, \ldots, u_n \neq \{v_1, \ldots, v_m\}$, $n \geqslant 2$, with the $u_i$ and $v_j$ of sort either *Set*, or $s$ or less; (iv) $u \neq \{v_1, \ldots, v_m\}$, with the $u$ of sort $s$ or less, and the $v_j$ of sort either *Set*, or $s$ or less; (v) $u \neq v_1, \ldots, v_m$, $(m \geqslant 2)$ with the $u$ of sort $s$ or less, and the $v_j$ of sort either *Set*, or $s$ or less; (vi) $\{u_1, \ldots, u_n\} \neq \{v_1, \ldots, v_m\}$, with the $u_i$ and $v_j$ of sort either *Set*, or $s$ or less; and (vii) $u_1, \ldots, u_n \neq v_1, \ldots, v_m$, $n, m \geqslant 2$, with with the $u_i$ and $v_j$ of sort either *Set*, or $s$ or less.

Case (i) with $n = 1$ is a valid disequality and can be ignored; and for $n > 1$ can, by normalization, be replaced by the conjunction $\bigwedge_{i \neq i'} u_i \neq u_{i'}$. Cases (ii) and (iv) are *valid* disequalities in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$ and therefore can be ignored in $\bigwedge D'$. This leaves us with cases (iii) and (v)–(vii). In case (iii), by normalization, we must have $u_i \neq_{AC \cup B_{(\Omega',\Delta')}} u'_i$ when $i \neq i'$, and we can replace the disequality by the conjunction $\bigwedge_{i \neq i'} u_i \neq u'_i$. In case (v), by normalization,

we must have $v_j \neq_{AC \cup B_{(\Omega',\Delta')}} v'_j$ for $j \neq j'$, and we can replace that disequality by the conjunction $\bigwedge_{j \neq j'} v_j \neq v_{j'}$. In cases (vi)–(vii), by normalization, we must have $u_i \neq_{AC \cup B_{(\Omega',\Delta')}} u'_i$ when $i \neq i'$, and $v_i \neq_{AC \cup B_{(\Omega',\Delta')}} v'_j$ when $j \neq j'$. And in both cases, if $n \neq m$, we can replace (vi)–(vii) by the conjunction $\bigwedge_{i \neq i'} u_i \neq u'_i \wedge \bigwedge_{j \neq j''} v_i \neq v'_j$. If $n = m$, in both cases normalization and axiom-consistency force the existence of a $u_q$ such that $u_q \neq_{AC \cup B_{(\Omega',\Delta')}} v_i$, $1 \leqslant i \leqslant n$, and we can replace (vi)–(vii) by the conjunction $\bigwedge_{1 \leqslant i \leqslant n} u_q \neq v_i$.

This leaves us with predicate disequalities $u \subseteq v \neq tt$, which can be of one of the following forms: (i) $\{\overline{u}_k\} \subseteq \varnothing \neq tt$, $k \geqslant 1$, with the $u_i$ terms of sort *Set* or $s$ or less, which is a *valid* disequality in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$ and therefore can be ignored in $\bigwedge D'$, and (ii) (up to $AC \cup B_{(\Omega',\Delta')}$-equality), $\{\overline{u}_k, \overline{v}_n\} \subseteq \{\overline{u}_k, \overline{w}_m\} \neq tt$, with $k \geqslant 0$, $n \geqslant 1$, and if $k = 0$ then $m \geqslant 1$, where all individual terms in $\overline{u}_k, \overline{v}_n, \overline{w}_m$ are mutually $AC \cup B_{(\Omega',\Delta')}$-different terms of sort *Set* or $s$ or less. Therefore, we can replace this disequality by the conjunction, $\bigwedge_{1 \leqslant i \leqslant k} v_1 \neq u_i \wedge \bigwedge_{1 \leqslant j \leqslant m} v_1 \neq w_j$, where one of the two conjuncts may possibly be absent.

In this way we obtain our desired axiom-consistent and normalized conjunction $\bigwedge D''$ of $(\Omega \cup \Omega', \Pi \cup \Delta')$-disequalities whose variables have infinite sorts, that *is* satisfiable in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$ by the induction hypothesis, and whose satisfiability implies that of $\bigwedge D'$. In particular this proves that $\bigwedge D\{\overline{Y} \mapsto \{\overline{y}\}\}$ is satisfiable in $C_{\mathcal{H}_{(\Omega,\Pi)}[\mathcal{G}_{(\Omega',\Delta')},X \mapsto s]}$, as desired. □

By the above theorem, validity of all QF inductive theorems in an instance of the HF sets module satisfying the requirements in the theorem is decidable. Therefore, we can decide, for example, that $C_{\mathcal{H}[\mathcal{G},X \mapsto s]}$ satisfies theorems such as: the *extensionality* axiom $(U \subseteq V \wedge V \subseteq U) \Rightarrow U = V$, the *pairing axiom*, $x \in \{S, S'\} \Leftrightarrow (x \in S \vee x \in S')$, the *extensionality of ordered pairs lemma*, $\{x, \{x, y\}\} = \{x', \{x', y'\}\} \Rightarrow (x = x' \wedge y = y')$, the *finite union axiom*, $x \in (S \cup S') \Leftrightarrow (x \in S \vee x \in S')$, the equivalence $x \in S = tt \Leftrightarrow S = (S \cup \{x\})$, the *associativity-commutativity and idempotency* of $\cup$, and so on.

Let me do in detail the extensionality of ordered pairs lemma (which holds of course for all instances) for the instance $\mathcal{H}[\mathcal{N}_+, X \mapsto Nat]$. Proving this is equivalent to checking the unsatisfiability in $C_{\mathcal{H}[\mathcal{N}_+,X \mapsto Nat]}$ of the two conjunctions: $\{x, \{x, y\}\} = \{x', \{x', y'\}\} \wedge x \neq x'$, and $\{x, \{x, y\}\} = \{x', \{x', y'\}\} \wedge y \neq y'$. The equation $\{x, \{x, y\}\} = \{x', \{x', y'\}\}$ has the single, variant-based, unifier: $\{x \mapsto x', y \mapsto y'\}$, yielding the unsatisfiable formulas $x' \neq x'$, and $y' \neq y'$, as desired.

*Remark 1.* The standard HF sets without "urelements" can be seen as the instance of $\mathcal{H}[X]$ where $X$ is instantiated to an empty sort in a single-sorted theory, which, since sorts are always assumed non-empty, falls outside the conditions in the above theorem. However, the above proof of compactness can be adapted to the case of standard HF sets as follows. We first remove the parameter sort $X$ and the sort *Elt* and make the module unparameterized. All rules remain the same, but the $\in$ predicate is now typed $\_ \in \_ : \mathit{Set}\ \mathit{Set} \rightarrow \mathit{Pred}$, so the variable $x$ of sort *Elt* should now have sort *Set*. All sorts are infinite. Given a reduced and $AC$-consistent conjunction of disequalities $\bigwedge D$, to prove that it is satisfiable in

the initial model, if $\overline{Y}$ are its variables, of sort either *Magma* or *Set*, and all syntactically different, we generate fresh new variables $\overline{y}$ of sort *Set* corresponding to the $\overline{Y}$, and instantiate $\bigwedge D$ by the substitution $\{\overline{Y} \mapsto \{\overline{y}\}\}$. We then define $\mu$ as before and prove satisfiability of conjunctions like $\bigwedge D\{\overline{Y} \mapsto \{\overline{y}\}\}$ by induction on $\mu$.

## 8 Descent Maps and QF-Decidable Cores

Sections 5–7 have given two general methods —one through the Descent Theorems and Corollary 2, and another through parameterization— to substantially enlarge the class of OS-FO equational theories for which satisfiability of QF formulas in their initial model is decidable. This section proposes a third method that will further enlarge such a class and that includes the first two methods as special cases. When applying the methods and the generic satisfiability algorithms developed so far we may run into both theoretical and practical barriers:

1. At the theoretical level, an FVP theory decomposition $\mathcal{R} = (\Sigma \cup \Pi, B, R)$ of an OS-FO equational theory $((\Sigma, \Pi), \Gamma)$ may protect a constructor decomposition $\mathcal{R}_{(\Omega,\Delta)}$ that is *not* OS-compact, so the methods developed so far cannot be applied (see the positively defined Presburger arithmetic below).
2. At the practical level we may run into serious *performance barriers* when applying the generic satisfiability checking algorithm to a given FVP decomposition $\mathcal{R}$, particularly when $\mathcal{R}$ has a relatively high variant complexity on top of a unification algorithm with high computational complexity.

What can be done? Both problems can be addressed —thus enlarging both the theoretical and practical reach of variant-based satisfiability— by means of what I call *descent maps*. The idea generalizes that of the algorithm "unpacking" Corollary 2, where we mapped a DNF formula $\varphi = \bigvee_i \bigwedge G_i \wedge \bigwedge D_i$ in the OS-FO theory decomposed by $\mathcal{R}$ into an equi-satisfiable $(\Omega, \Delta)$-formula $\varphi^\delta = \bigvee_{i,\beta,j} \bigwedge D_i^j \alpha$, —where $\alpha$ ranges over the constructor unifiers of $\bigwedge G_i$, and $j$ over the axiom-consistent $(\Omega, \Delta)$-variants of $\bigwedge D_i\alpha$— for the constructor decomposition $\mathcal{R}_{(\Omega,\Delta)}$. The generalization is twofold: (i) instead of a constructor decomposition we allow any decomposition $\mathcal{D}$ which is *conservatively extended* by $\mathcal{R}$ (recall Definition 4), and (ii) instead of the mapping $\varphi \mapsto \varphi^\delta$ we allow any user-definable mapping $\varphi \mapsto \varphi^\bullet$ such that $\varphi$ is satisfiable in $C_\mathcal{R}$ iff $\varphi^\bullet$ is satisfiable in $C_\mathcal{D}$. Here is the precise definition:

**Definition 8.** *A* descent map *is a triple $(\mathcal{R}, \bullet, \mathcal{D})$ where $\mathcal{R}$ and $\mathcal{D}$ are decompositions of equational OS-FO theories, and $\mathcal{R}$ conservatively extends $\mathcal{D}$, and where $\bullet$ is a total*[19] *computable function, $\varphi \mapsto \varphi^\bullet$, mapping each QF-FO formula $\varphi$ in the theory decomposed by $\mathcal{R}$ into a corresponding QF-FO formula $\varphi^\bullet$*

[19] This requirement can be relaxed by defining a descent map as a triple $((\mathcal{R}, \Gamma), \bullet, (\mathcal{D}, \Gamma'))$, with $\Gamma$ and $\Gamma'$ sets of QF formulas in their respective theories, $\Gamma$ the domain of $\bullet$, and $\Gamma^\bullet \subseteq \Gamma'$. This may be useful because sometimes mappings assume prior mappings putting formulas in a particular shape. Another generalization

*in the theory decomposed by $\mathcal{D}$ and such that $C_{\mathcal{R}} \models \exists\, \varphi \;\Leftrightarrow\; C_{\mathcal{D}} \models \exists\, \varphi^{\bullet}$, where $\exists\, \varphi$ denotes the existential closure of $\varphi$.*

*We say that $(\mathcal{R}, \bullet, \mathcal{D})$, and $\mathcal{R}$, have $\mathcal{D}$ as a* QF-decidable core*, or just a* core*, if satisfiability in $C_{\mathcal{D}}$ of any QF-FO formula in the theory decomposed by $\mathcal{D}$ is decidable. If, in addition, $\mathcal{D}$ is OS-compact, then we call $\mathcal{D}$ an* OS-compact core*.*

The importance of $\mathcal{R}$ having a core is that this automatically makes the satisfiability of QF formulas in $C_{\mathcal{R}}$ decidable. An important feature of descent maps is their *compositionality*. That is, if $(\mathcal{R}, \bullet, \mathcal{D})$ and $(\mathcal{D}, \diamond, \mathcal{Q})$ are descent maps, then $(\mathcal{R}, \bullet\diamond, \mathcal{Q})$ (note the *diagrammatic* order of function composition) is also a descent map. This is so because conservative extension inclusions and computable maps compose, and because of the equivalences:

$$C_{\mathcal{R}} \models \exists\, \varphi \;\Leftrightarrow\; C_{\mathcal{D}} \models \exists\, \varphi^{\bullet} \;\Leftrightarrow\; C_{\mathcal{Q}} \models \exists\, \varphi^{\bullet\diamond}$$

This can be very useful, because: (i) we do not need to reach a core by means of a single map: we may do so after several steps of composition; and (ii) we can *reuse* specific descent maps by composing them in various ways with other such maps. An enlightened way to think about descent maps and understand their compositional structure is to realize that they form a *category*, whose objects are decompositions[20] and whose arrows are the computable functions $\bullet$, $\diamond$, and so on. From now on I will write a descent map $(\mathcal{R}, \bullet, \mathcal{D})$ as a morphism $\mathcal{R} \xrightarrow{\bullet} \mathcal{D}$. This categorical structure has useful consequences: for $\mathcal{D}$ to be a core of $\mathcal{R} \xrightarrow{\bullet} \mathcal{D}$ it is enough to have a descent map $\mathcal{D} \xrightarrow{\diamond} \mathcal{G}$ with $\mathcal{G}$ a core of $\mathcal{D}$; and then both $\mathcal{G}$ and $\mathcal{D}$ are cores of $\mathcal{R}$, but $\mathcal{G}$, being smaller, may be a better core than $\mathcal{D}$.

I will illustrate the usefulness of descent maps with two running examples, where Presburger arithmetic on the naturals, resp. the integers, is defined in a *positive* way. That is, the predicates $>$ and $\geqslant$ are defined as functions of sort *Pred* by giving equations characterizing only the cases when $t > t' = tt$ (resp. $t \geqslant t' = tt$) holds in the initial algebra. The *negative* cases when these predicates fail to hold do not have to be defined explicitly, because they are exactly the cases when $t > t' \neq tt$ (resp. $t \geqslant t' \neq tt$). There are three reasons why these positive presentations of Presburger arithmetic seem worth discussing:

1. They are simpler and have lower variant complexity than their counterparts with Boolean-valued predicates in Section 6.2, which makes them better candidates for an implementation.
2. They do *not* have compact constructor decompositions, so that the methods developed so far do not apply to them. Therefore, proving that they can be used for checking satisfiability of Presburger arithmetic QF-formulas both requires and illustrates the methods of this section.

---

is allowing quantified formulas in $\Gamma$ and $\Gamma'$. In this way, descent maps associated with *quantifier elimination procedures* can also be included in the framework. For further generalizations, see the remarks at the end of this section.

[20] More properly, pairs $(\mathcal{R}, ((\Sigma, \Pi), \Gamma))$, with $\mathcal{R}$ a decomposition of $((\Sigma, \Pi), \Gamma))$, but I will avoid notational purism and leave $((\Sigma, \Pi), \Gamma))$ implicit.

3. They illustrate a much more general dilemma between predicates, the finite variant property, and OS-compactness. The dilemma is that, on the one hand, it is often much easier to get an FVP specification if predicates are defined only in the positive (true) case. Think, for example, about the membership and containment predicates for sets or HF sets, which were only defined in the positive case for exactly this reason. On the other hand, unless a predicate $p$ is always true, it is a constructor that will *never* be free unless always false, so that the standard methods we have currently available to show OS-compactness do not apply. Note that: (i) several proofs of decidable satisfiability for parameterized data types in Section 7 had to grapple with the problem of such positively-defined predicates to show we could reach an OS-compact constructor decomposition; and (ii) as pointed out in (2) above, such OS-compact constructor decomposition may not exist in general, yet decidable satisfiability may still be achieved by other means.

These examples will also show that, what I call descent map extensions, provide a *seamless combination algorithm* for deciding satisfiability in initial models of combined theories when these positive definitions of natural and integer *Presburger arithmetic* are combined within larger FVP specifications extending the Presburger arithmetic ones, under fairly mild assumptions about such larger specifications. By "seamless" I mean that *no combination infrastructure* à la Nelson-Oppen (NO) [79,82] is needed. Instead, a NO combination builds —and has to work at non-trivial computational cost through the "seams" of— the infrastructure needed to put the various theories together.

*Example 22.* (Positive Definition of Natural Presburger Arithmetic). An FVP decomposition $\mathcal{N}_{+,>,\geqslant}$ having the natural numbers with $+$, $>$ and $\geqslant$ as its initial model is obtained by a very simple extension of the FVP decomposition $\mathcal{N}_{+}$ in Footnote 7: we just add a sort *Pred* and constructor predicates $\_ > \_ : Nat\ Nat \rightarrow Pred$ defined by the rules $p+n > n \rightarrow tt$ and $n+m \geqslant n \rightarrow tt$, where $p$ is a variable of sort *NzNat* and $n, m$ variables of sort *Nat*. This yields an FVP decomposition with variant complexity 4.

Note that $\mathcal{N}_{+,>,\geqslant}$ is made up entirely of constructors, so $\mathcal{N}_{+,>,\geqslant}$ is its own constructor decomposition. However, $\mathcal{N}_{+,>,\geqslant}$ is *not* OS-compact, since the negation of the trichotomy law $n > m \vee m > n \vee n = m$ is the $AC$-consistent but *unsatisfiable* conjunction of disequalities $n > m \neq tt \wedge m > n \neq tt \wedge n \neq m$. However, I show in what follows that $\mathcal{N}_{+,>,\geqslant}$ has $\mathcal{N}_{+}$ as its OS-compact core. In fact, I show this, and more, in a compositional way: given an FVP $\mathcal{R}$ extending $\mathcal{N}_{+,>,\geqslant}$, I show that $\mathcal{R}$ itself also has a core, under mild assumptions on $\mathcal{R}$.

Before getting on with the details I need to explain the notion of a descent map *extension*, which will be key to articulate more precisely the seamless way in which Presburger arithmetic is combined with a decision procedure for the rest of a larger FVP specification $\mathcal{R}$. Here is the definition:

**Definition 9.** *An* extension *of a descent map $\mathcal{R}_0 \xrightarrow{\bullet} \mathcal{D}_0$ is another descent map $\mathcal{R} \xrightarrow{\bullet} \mathcal{D}$ such that: (i) $\mathcal{R} \supseteq \mathcal{R}_0$ and $\mathcal{D} \supseteq \mathcal{D}_0$ are protecting extensions, and (ii) the*

*function* $\bullet$ *on the QF formulas of the theory decomposed by* $\mathcal{R}_0$ *is a restriction of the function* $\bullet$ *on the QF formulas of the theory decomposed by* $\mathcal{R}$.

I show in what follows that descent map extensions play an important role in making satisfiability decision procedures *usable* in much broader contexts than their original ones, and can do so in a seamless way. Note that having an extension means that the descent map is extended to a *richer language of formulas*. Since many descent maps can be defined by (metalevel) rewrite rules which automatically apply to terms in a richer signature, the definition of extended descent maps will be straightforward in all examples I will present. However, the delicate part that still needs to be checked in each case is that the extended map so defined meets the requirements of a descent map.

I will first define a descent map $\mathcal{N}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{N}_+$ bringing $\mathcal{N}_{+,>,\geqslant}$ into the compact core $\mathcal{N}_+$. This map is itself the composition of two simpler ones *lit2at* and $\delta_0$. Then I will show how it has an extension to any larger FVP context $\mathcal{R}$ extending $\mathcal{N}_{+,>,\geqslant}$, under mild assumptions on $\mathcal{R}$. I will then illustrate the usefulness of this combination result with examples.

Let me first explain the descent map $\mathcal{N}_{+,>,\geqslant} \overset{lit2at}{\rightarrow} \mathcal{N}_{+,>,\geqslant}$. Its purpose is to eliminate all negated atoms of sort *Pred*, i.e., negated atoms of the form $u > v \neq tt$, resp. $u \geqslant v \neq tt$. The descent map *lit2at* (literal to atom) assumes formulas $\varphi$ in DNF and uses the theorem $n \geqslant m \Leftrightarrow \neg(m > n)$ of natural Presburger arithmetic discussed in Example 11 to replace each literal of the form $u > v \neq tt$ (resp. $u \geqslant v \neq tt$) in $\varphi$ by the atom $v \geqslant u = tt$ (resp. $v > u = tt$). The second descent map $\mathcal{N}_{+,>,\geqslant} \overset{\delta_0}{\rightarrow} \mathcal{N}_{+,>,\geqslant}$ transforms each conjunction $\bigwedge G \wedge \bigwedge D$ in a DNF formula $\varphi$ into the disjunction $\bigvee_{\alpha \in Unif_E(\bigwedge G)} \bigwedge D\alpha$, where $E$ are the equations in the theory decomposed by $\mathcal{N}_{+,>,\geqslant}$, so $\delta_0$ preserves satisfiability.

The reason why we can type the composed map as $\mathcal{N}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{N}_+$ is the following. Each conjunction $\bigwedge G \wedge \bigwedge D$ of equations $G$ and disequations $D$ in natural Presburger arithmetic decomposes, taking account of sorts in disequalities, as $\bigwedge G \wedge \bigwedge D_{Nat} \wedge \bigwedge D_{Pred}$. But *lit2at* will transform $\bigwedge D_{Pred}$ into a conjunction of equalities $\bigwedge G'$. And then $\delta_0$ will transform $\bigwedge G \wedge \bigwedge G' \wedge \bigwedge D_{Nat}$ into the disjunction $\bigvee_{\alpha \in Unif_E(\bigwedge G \wedge \bigwedge G')} \bigwedge D_{Nat}\alpha$, which is a formula in $\mathcal{N}_+$.

This automatically provides a simple decision procedure for the *positive* definition of natural Presburger arithmetic. Let me illustrate how this works by proving the transitivity law $(n > m = tt \wedge m > n' = tt) \Rightarrow n > n' = tt$ of Presburger arithmetic. Its negation is the conjunction $n > m = tt \wedge m > n' = tt \wedge n > n' \neq tt$, which is mapped by *lit2at* to the conjunction $n > m = tt \wedge m > n' = tt \wedge n' \geqslant n = tt$, which is mapped by $\delta_0$ to the empty disjunction (false), because $n > m = tt \wedge m > n' = tt \wedge n' \geqslant n = tt$ has no variant unifiers.

Let me now show how the above descent-based decision procedure for the positive definition of natural Presburger arithmetic can be seamlessly combined with a decision procedure for the rest of any reasonable FVP decomposition $\mathcal{R}$ extending $\mathcal{N}_{+,>,\geqslant}$. By a "reasonable extension" I mean something quite specific: the notion of a *proper extension*, which I first define and then justify.

**Definition 10.** *Let $\mathcal{R}_0 \subseteq \mathcal{R}$ be an inclusion of decompositions $\mathcal{R}_0 = (\Sigma_0 \cup \Pi_0, B_0, R_{\Sigma_0} \cup R_{\Pi_0})$ and $\mathcal{R} = (\Sigma \cup \Pi, B, R_\Sigma \cup R_\Pi)$ of OS-FO theories, so that $R_\Sigma$ (resp. $R_{\Sigma_0}$) are the rules defining function symbols in $\Sigma$ (resp. $\Sigma_0$), and $R_\Pi$ (resp. $R_{\Pi_0}$) are the rules defining predicates in $\Pi$ (resp. $\Pi_0$). Call $\mathcal{R}_0 \subseteq \mathcal{R}$ a* proper extension *iff: (i) $\mathcal{R}_{\Sigma_0} \subseteq \mathcal{R}_\Sigma$ is a protecting extension, where $\mathcal{R}_{\Sigma_0} = (\Sigma_0, B_0, R_{\Sigma_0})$ and $\mathcal{R}_\Sigma = (\Sigma, B, R_\Sigma)$, (i.e., all predicates are removed); and (ii) the signature $\Pi_1 = \Pi - \Pi_0$ consists exclusively of new predicate symbols outside $\Pi_0$ (i.e., no subsort overloading exists between predicates in $\Pi_0$ and in $\Pi_1$), and the rules in $R_\Pi - R_{\Pi_0}$ only define predicates in $\Pi_1$ and make no use of symbols in $\Pi_0$, abbreviated hereafter as $R_\Pi = R_{\Pi_0} \uplus R_{\Pi_1}$.*

Intuitively, $\mathcal{R}_0 \subseteq \mathcal{R}$ proper means that we extend $\mathcal{R}_0$ by possibly adding new data and functions, but protecting the previous ones, and by possibly adding totally new predicates $\Pi_1$, that are defined independently of the predicates $\Pi_0$. This is, I believe, a quite "reasonable" notion of extension frequently used in practice. Proper extensions enjoy several good properties:

**Lemma 2.** *Let $\mathcal{R}_0 \subseteq \mathcal{R}$ be a proper extension. Then:*

1. *For any first-order formula $\varphi$ in the theory decomposed by $\mathcal{R}_0$ we have*

$$C_\mathcal{R} \models \exists\, \widetilde{\varphi} \;\Leftrightarrow\; C_{\mathcal{R}_0} \models \exists\, \widetilde{\varphi}.$$

2. *If $\mathcal{Q}_0 \subseteq \mathcal{R}_0$ is a proper extension with $\mathcal{Q}_0 = (\Sigma_0 \cup \Delta_0, B_0, R_{\Sigma_0} \cup R_{\Delta_0})$ (i.e., $\mathcal{Q}_0$ has the same data and functions as $\mathcal{R}_0$, and predicates in a predicate subsignature $\Delta_0 \subseteq \Pi_0$) then $\mathcal{Q}_0 \subseteq \mathcal{Q}$ and $\mathcal{Q} \subseteq \mathcal{R}$ are also proper extensions, where $\mathcal{Q} = (\Sigma_0 \cup \Delta_0 \cup \Pi_1, B_0, R_{\Sigma_0} \cup R_{\Delta_0} \cup R_{\Pi_1})$. Furthermore, if $\mathcal{R}$ is FVP, so is $\mathcal{Q}$.*

*Proof.* (1) follows easily from $\mathcal{R}_{\Sigma_0} \subseteq \mathcal{R}_\Sigma$ being a protecting extension and the fact that $R_\Pi = R_{\Pi_0} \uplus R_{\Pi_1}$. This means that ground equalities and ground $\Pi_0$-atoms are evaluated in the exact same way in $C_\mathcal{R}$ and $C_{\mathcal{R}_0}$ and, furthermore, that the sets of assignments in $C_\mathcal{R}$ and $C_{\mathcal{R}_0}$ for the variables $\mathit{fvars}(\widetilde{\varphi})$ *coincide*.

To prove (2) we first of all need to check that $\mathcal{Q}$ is indeed a decomposition. Since $\mathcal{Q}$ has possibly fewer rules than $\mathcal{R}$, termination is trivial. For $\Sigma$-terms the rewrite relations in $\mathcal{Q}$ and $\mathcal{R}$ coincide, and, by $\mathcal{R}_0 \subseteq \mathcal{R}$ and $\mathcal{Q}_0 \subseteq \mathcal{R}_0$ being proper extensions, we have $R_\Pi = (R_{\Delta_0} \cup R_{\Pi_1}) \uplus R_{\Pi - (\Delta_0 \cup \Pi_1)}$, so that the rewrite relations of $\mathcal{Q}$ and $\mathcal{R}$ agree on $(\Delta_0 \cup \Pi_1)$-predicate terms. This ensures confluence as well. Since the rewrite relations agree on all common terms, if $\mathcal{R}$ is FVP, so is $\mathcal{Q}$. The easy check that $\mathcal{Q}_0 \subseteq \mathcal{Q}$ and $\mathcal{Q} \subseteq \mathcal{R}$ are proper extensions is left to the reader. □

**Theorem 14.** *(Combination Theorem for Positively Defined Natural Presburger Arithmetic). Let $\mathcal{N}_{+,>,\geqslant} \subseteq \mathcal{R}$ be a proper extension with $\mathcal{R}$ FVP. Let then $\mathcal{N}_+ \subseteq \mathcal{Q} \subseteq \mathcal{R}$ be the proper extensions associated by (2) in Lemma 2 to the proper extensions $\mathcal{N}_+ \subseteq \mathcal{N}_{+,>,\geqslant} \subseteq \mathcal{R}$, which ensures $\mathcal{Q}$ is also FVP. Then, then descent map $\mathcal{N}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{N}_+$ has an extension to a descent map $\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q}$. Therefore, if $\mathcal{Q}$ has a core, $\mathcal{R}$ also has a core, so that satisfiability of QF formulas in the initial algebra of $C_\mathcal{R}$ becomes decidable.*

*Proof.* The descent map $\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q}$ is define exactly as $\mathcal{N}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{N}_{+}$. That is, as a composition of $\mathcal{R} \overset{lit2at}{\longrightarrow} \mathcal{R}$ with $\delta_0$, where, as before, $\mathcal{R} \overset{lit2at}{\longrightarrow} \mathcal{R}$ replaces literals of the form $u > v \neq tt$ (resp. $u \geqslant v \neq tt$) in a DNF formula $\widetilde{\varphi}$ of the OS-FO theory decomposed by $\mathcal{R}$ by atoms $v \geqslant u = tt$ (resp. $v > u = tt$). Therefore, since $lit2at\,\delta_0$ removes all occurrences of literals for the predicates $>$ and $\geqslant$ in $\widetilde{\varphi}$, and $\mathcal{Q}$ is obtained from $\mathcal{R}$ precisely by removing $>$ and $\geqslant$, the typing $\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q}$ makes sense. Furthermore, the equivalence $C_{\mathcal{R}} \models \exists\, \widetilde{\varphi}^{lit2at} \Leftrightarrow C_{\mathcal{Q}} \models \exists\, \widetilde{\varphi}^{lit2at\,\delta_0}$ follows from (1) in Lemma 2 applied to the proper extension $\mathcal{Q} \subseteq \mathcal{R}$. Therefore, all we are left to do is to prove that $\mathcal{R} \overset{lit2at}{\longrightarrow} \mathcal{R}$ is indeed a descent map, i.e., need to show the equivalence $C_{\mathcal{R}} \models \exists\, \widetilde{\varphi} \Leftrightarrow C_{\mathcal{R}} \models \exists\, \widetilde{\varphi}^{lit2at}$. But this follows from the fact that, by Lemma 2 applied to the proper extension $\mathcal{N}_{+,>,\geqslant} \subseteq \mathcal{R}$, we have $C_{\mathcal{R}} \models (x \geqslant y \Leftrightarrow \neg(y > x))$, ensuring that $\mathcal{R} \overset{lit2at}{\longrightarrow} \mathcal{R}$ is a descent map. □

One could politely ask: where is the so-called "seamlessness" of the combination algorithm or, for that matter, the algorithm itself to be found? Well, it is all there, we just need to "unpack" Theorem 14 a little. The theorem's formulation is very general. Since all we know is that $\mathcal{Q}$ has a core, we do not have a *fixed* algorithm for deciding the satisfiability of $\widetilde{\varphi}^{lit2at\,\delta_0}$, since further formula descent maps transforming $\widetilde{\varphi}^{lit2at\,\delta_0}$ may have to be applied. What we have is a composition of a *fixed* first step $\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q}$ with a *variable* second step. So let me also fix the second step by focusing on a very common scenario of use: suppose that $\mathcal{Q}$ has a OS-compact constructor decomposition $\mathcal{Q}_{(\Omega,\Delta)}$. Then the satisfiability checking algorithm is given by the composed descent map

$$\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q} \overset{\delta}{\longrightarrow} \mathcal{Q}_{(\Omega,\Delta)}$$

But the second step, $\delta$, in this composed descent map just encapsulates the *generic algorithm* reducing an FVP theory to its OS-compact constructor decomposition described after Corollary 2. That is, checking the satisfiability of $\widetilde{\varphi}^{lit2at\,\delta_0}$ is just the last check explained after Corollary 2. So the *seamlessness* of the combination boils down to the fact that the simple first descent map $\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q}$ *takes care of all Presburger arithmetic matters uniformly, for any extended context* $\mathcal{R}$, and the second step is just business as usual: no abstraction of variables, and no Nelson-Oppen-like comings and goings across variables shared between $\mathcal{N}_{+,>,\geqslant}$ and the rest of $\mathcal{R}$ are needed at all.

All this can be further illustrated by reinterpreting all of Section 7 in terms of extensions of descent maps. Specifically, Theorems 9–13 can all be summarized by saying that, under suitable assumptions on a chosen sort $s$ in an FVP decomposition $\mathcal{G}$ with an OS-compact constructor decomposition $\mathcal{G}_{(\Omega',\Delta')}$, and for $\mathcal{R}[X]$ any of the parameterized data types mentioned in those theorems, the descent map $\mathcal{G} \overset{\delta}{\rightarrow} \mathcal{G}_{(\Omega',\Delta')}$ has an OS-compact-core-preserving extension to the descent map $\mathcal{R}[\mathcal{G}, X \mapsto s] \overset{\delta}{\rightarrow} \mathcal{R}_{(\Omega,\Delta)}[\mathcal{G}_{(\Omega',\Delta')}, X \mapsto s]$. But if in Theorem 14 we specialize its generic proper extension $\mathcal{R} \supseteq \mathcal{N}_{+,>,\geqslant}$ to

$\mathcal{R}[\mathcal{N}_{+,>,\geqslant}, X \mapsto Nat] \supseteq \mathcal{N}_{+,>,\geqslant}$, then $\mathcal{Q}$ specializes to $\mathcal{R}[\mathcal{N}_+, X \mapsto Nat]$ protecting $\mathcal{N}_+$. Therefore, the seamless combination algorithm of natural Presburger arithmetic for all the above parameterized dat types is just encapsulated in the composed descent map:

$$\mathcal{R}[\mathcal{N}_{+,>,\geqslant}, X \mapsto Nat] \xrightarrow{lit2at\,\delta_0} \mathcal{R}[\mathcal{N}_+, X \mapsto Nat] \xrightarrow{\delta} \mathcal{R}_{(\Omega,\Delta)}[\mathcal{N}_+, X \mapsto Nat].$$

Note, finally, that since the conditions in Theorems 9–13 still apply to any such $\mathcal{R}[\mathcal{N}_+, X \mapsto Nat] \xrightarrow{\delta} \mathcal{R}_{(\Omega,\Delta)}[\mathcal{N}_+, X \mapsto Nat]$ when interpreted as the *instance data type* $\mathcal{G} = \mathcal{R}[\mathcal{N}_+, X \mapsto Nat]$, the process can be *iterated* and applied to nested parameterized data types like sets of lists of naturals, HF sets of compact lists of multisets of naturals, and so on.

Consider, for example, the data type $\mathcal{H}[\mathcal{L}[\mathcal{N}_{+,>,\geqslant}, X \mapsto Nat], X \mapsto List]$ of HF sets, whose urelements are lists of (positively defined) Presburger natural numbers. To check the satisfiability of the formula

$$head(l) > head(l') = tt \;\wedge\; head(l) > 1 + 1 + 1 \neq tt \;\wedge\; \{(1 + 1); nil\} \subseteq \{l, l'\} \neq tt$$

with $l, l'$ of sort *NeList*, we first apply the *lit2at* descent map to get

$$head(l) > head(l') = tt \;\wedge\; 1 + 1 + 1 \geqslant head(l) = tt \;\wedge\; \{(1 + 1); nil\} \subseteq \{l, l'\} \neq tt$$

The system of equations $head(l) > head(l') = tt \;\wedge\; 1 + 1 + 1 \geqslant head(l) = tt$ has six variant unifiers. $\delta_0$ just applies them to the conjunction's disequality. One of the unifiers is $\{l \mapsto (1 + 1 + 1); l_1, l' \mapsto (1 + 1); l_2\}$, with $l_1, l_2$ of sort *List*, which instantiates the disequality as: $\{(1 + 1); nil\} \subseteq \{(1 + 1 + 1); l_1, (1 + 1); l_2\} \neq tt$, which is a normalized and *ACU*-consistent constructor disequality. Thus, the original formula is satisfiable.

Let me now specify Presburger arithmetic on the integers in a positive way as an FVP theory.

*Example 23.* (Positive Definition of Integer Presburger Arithmetic). The FVP theory $\mathcal{Z}_{+,>,\geqslant}$ of integer Presburger arithmetic protects $\mathcal{Z}_+$ by adding a sort *Pred* and two predicates $\_ > \_, \_ \geqslant \_ : Int\ Int \to Pred$ to its constructor signature and rules $p + n > n \to tt$, $n > -(q) \to tt$, $-(p) > -(p + q) \to tt$, for $>$, and rules $n + m \geqslant n \to tt$, $n \geqslant -(q) \to tt$, $-(p) \geqslant -(p) \to tt$, $-(p) \geqslant -(p + q) \to tt$ for $\geqslant$, were $p, q$ have sort *NzNat* and $n, m$ have sort *Nat*. This theory is FVP with variant complexity 21.

The same negated trichotomy law showing that the constructor decomposition of $\mathcal{N}_{+,>,\geqslant}$ is not OS-compact proves the same result for $\mathcal{Z}_{+,>}$, even when $n$ and $m$ remain of sort *Nat*, although here one would prefer to type $n$ and $m$ with sort *Int*.

Since the equivalence $i \geqslant j \Leftrightarrow \neg(j > i)$ discussed in Example 15 is a theorem of integer Presburger arithmetic, the same reasons proving the correctness of the descent map $\mathcal{N}_{+,>,\geqslant} \xrightarrow{lit2at\,\delta_0} \mathcal{N}_+$ prove also the correctness of an entirely similar descent map $\mathcal{Z}_{+,>,\geqslant} \xrightarrow{lit2at\,\delta_0} \mathcal{Z}_+$. Also, the Combination Theorem 14 for

positively defined natural Presburger arithmetic extends naturally to the Combination Theorem below for positively defined integer Presburger arithmetic. It is stated without proof because the proof arguments are, *mutatis mutandis*, exactly those already given in Theorem 14.

**Theorem 15.** *(Combination Theorem for Positively Defined Integer Presburger Arithmetic). Let $\mathcal{Z}_{+,>,\geqslant} \subseteq \mathcal{R}$ be a proper extension with $\mathcal{R}$ FVP. Let then $\mathcal{Z}_+ \subseteq \mathcal{Q} \subseteq \mathcal{R}$ be the proper extensions associated by (2) in Lemma 2 to the proper extensions $\mathcal{Z}_+ \subseteq \mathcal{Z}_{+,>,\geqslant} \subseteq \mathcal{R}$, which ensures $\mathcal{Q}$ is also FVP. Then, then descent map $\mathcal{Z}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Z}_+$ has an extension to a descent map $\mathcal{R} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Q}$. Therefore, if $\mathcal{Q}$ has a core, $\mathcal{R}$ also has a core, so that satisfiability of QF formulas in the initial algebra of $C_{\mathcal{R}}$ becomes decidable.*

At the beginning of this section I mentioned two problems motivating the need for, and usefulness of, descent maps: (1) $\mathcal{R}$ can have a constructor decomposition that is not OS-compact; and (2) even if $\mathcal{R}$ is FVP and does have an OS-compact constructor decomposition, we can run into performance barriers — for example when computing constructor unifiers or constructor variants— due to $\mathcal{R}$'s relatively high variant complexity. As the examples of natural and integer Presburger arithmetic make clear, a single descent map can both solve problem (1) and make substantial progress towards solving problem (2): the descent map $\mathcal{N}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{N}_+$ both brings positively defined Presburger natural arithmetic to an OS-compact core, and reduces variant complexity from 4 to 0. Likewise, the descent map $\mathcal{Z}_{+,>,\geqslant} \overset{lit2at\,\delta_0}{\longrightarrow} \mathcal{Z}_+$ both brings integer Presburger arithmetic to the $\mathcal{Z}_+$ core and reduces variant complexity from 21 to 12.

Many more performance improvements using descent maps are at hand. For example, using formula transformations similar to those sketched out in [21], Appendix D defines a descent map $\mathcal{Z}_+ \overset{v\,-}{\longrightarrow} \mathcal{N}_+$ reducing $\mathcal{Z}_+$'s relatively high variant complexity from 12 to 0. However, because of the sort inclusion $NzNat < Nat$, in $\mathcal{N}_+$ we still have to perform *order-sorted ACU*-unification, which adds extra computational cost to unsorted *ACU*-unification. To avoid that extra cost, we can use a second descent map $\mathcal{N}_+ \overset{u}{\longrightarrow} \mathcal{N}_+^u$ to the unsorted theory $\mathcal{N}_+^u$ of Example 8, where —assuming that each variable name in a formula $\varphi$ has a unique sort— $u(\varphi)$ is the instantiation of $\varphi$ that leaves all variables of sort *Nat* unchanged and replaces each variable $x$ of sort *NzNat* in $\varphi$ by the term $x + 1$, where $x$ now has sort *Nat*.

Other performance-improving descent maps keep the theory unchanged and act only at the formula level. Notice, for example, that any group, or any free monoid (commutative or not), satisfies the cancellation equivalence: $x + y = x + z \Leftrightarrow y = z$. This means, for example, that in $\mathcal{M}[X]$, $\mathcal{Z}_+$, and even $\mathcal{N}_+$, we can use cancellation rewrite rules of the form: $M, M' = M, M'' \rightarrow M' = M''$, and $x + y = x + z \rightarrow y = z$, where, $M, M', M''$ have sort *MSet*, but —to avoid non-termination issues due to the *ACU* axioms— in $\mathcal{Z}_+$ $x$ must have sort either *NzNat* or *NzNeg*, and in $\mathcal{N}_+$ $x$ should have sort *NzNat*. This can be used to define descent maps $\mathcal{M}[X] \overset{cancel}{\longrightarrow} \mathcal{M}[X]$, $\mathcal{Z}_+ \overset{cancel}{\longrightarrow} \mathcal{Z}_+$, and

$\mathcal{N}_+ \stackrel{cancel}{\longrightarrow} \mathcal{N}_+$, that repeatedly apply the above rewrite rules to formulas to yield obviously equi-satisfiable but potentially much simpler formulas, which may require considerably less costly variant computations.

Similar cancellation equivalences, $x; l = y; l \Leftrightarrow x = y$ and $x; l = x; l' \Leftrightarrow l = l'$, hold for the parameterized list module $\mathcal{L}[X]$. They can be used as rewrite rules $x; l = y; l \rightarrow x = y$ and $x; l = x; l' \rightarrow l = l'$, to define a descent map $\mathcal{L}[X] \stackrel{cancel}{\longrightarrow} \mathcal{L}[X]$, that will likewise simplify list formulas and improve the efficiency of their variant-based computations.

The moral of this section is that we should think of the category of descent maps as a flexible, compositional *semantic framework* for satisfiability, where formula transformations (including quantifier elimination: see Footnote 19) and descent to simpler theories can be combined to both design new satisfiability algorithms and to improve the efficiency of existing ones, like those of the form $\mathcal{R} \stackrel{\delta}{\longrightarrow} \mathcal{R}_{(\Omega,\Delta)}$, which are automatically provided by the framework when $\mathcal{R}_{(\Omega,\Delta)}$ is an OS-compact constructor decomposition of an FVP $\mathcal{R}$. This has two useful consequences: (i) the compositional structure of descent maps can be used to give modular, more easily understandable, and often reusable proofs for the correctness of satisfiability algorithms; and (ii) performance problems can be dealt with by means of descent maps and, although part and parcel of prototyping a new satisfiability algorithm, they could be greatly reduced in the algorithm's optimized form.

Of course, nothing forces the optimized form of a satisfiability algorithm to be variant-based: it could be so, but need not be so: the notion of descent map is very general and is *independent* from the notion of FVP decomposition, so that it can be used to modularize and prove the correctness of satisfiability algorithms in general. This extra generality applies not just to equational OS-FO theories, but also to general theories and more general relations between theories and between formulas. Indeed, one should broaden the notion of descent map to allow general descent maps of the form $T \stackrel{(H,\bullet)}{\rightarrow} T'$, where $T$ and $T'$ are arbitrary theories,[21] $H : T' \rightarrow T$ is a theory *interpretation*, instead of just a theory inclusion $T' \subseteq T$, and $\bullet$ is generalized from being a function to being a relation between formulas that ensures equi-satisfiability across the classes of models of such theories.[22]

[21] $T$ and $T'$ can be first-order theories or, more generally —as it is commonly done in recent approaches to satisfiability— pairs $((\Sigma, \Pi), \mathcal{C})$ à la Ganzinger [49], where $\mathcal{C}$ is a class of $(\Sigma, \Pi)$-models. Indeed, one should think of satisfiability in initial models of equational OS-FO theories $((\Sigma, \Pi), \Gamma)$ as satisfiability for "theories" of the form: $((\Sigma, \Pi), [T_{\Sigma,\Pi,\Gamma}]_{\cong})$, where $[T_{\Sigma,\Pi,\Gamma}]_{\cong}$ denotes the *equivalence class* (class also in the *set theoretic* sense) of all models isomorphic to $T_{\Sigma,\Pi,\Gamma}$.

[22] That is, $\varphi \bullet \psi$ implies that $\varphi$ is satisfiable in $Mod(T)$ iff $\psi$ is satisfiable in $Mod(T')$. Examples of more general descent maps of this kind are provided by the reductions of the theory of *order-sorted* uninterpreted function symbols (resp. order-sorted function symbols modulo $AC$) to that of *unsorted* uninterpreted function symbols (resp. unsorted function symbols modulo $AC$) proved in [74]. This is achieved by descent maps $(\Sigma^u, \varnothing) \stackrel{(u,u^{-1})}{\longrightarrow} (\Sigma, \varnothing)$ (resp. $(\Sigma^u, AC^u) \stackrel{(u,u^{-1})}{\longrightarrow} (\Sigma, AC)$), where $\Sigma$ is an order-sorted signature, $\Sigma^u$ is the unsorted theory obtained from $\Sigma$ by identifying all

## 9 Related Work

The original paper proposing the concepts of variant and FVP is [32]. These ideas have been further advanced in [43,25,19,24]. In particular, I have used the ideas on folding variant narrowing and variant-based unification from [43], and have provided a different, detailed description of variant-based unifiers in Theorem 4 needed to better clarify the notion of constructor unifier in Section 4. To the best of my knowledge the notions of constructor variant and constructor unifier and the results on satisfiability in FVP initial algebras are new.

There is a vast literature on satisfiability in data types, including parameterized ones such as, e.g., [80,89,12,21,66,35,34]. In relation to that large body of work, what the results in this paper provide is both the characterization of a wide class of data types for which satisfiability is decidable, and a new *generic algorithm* to check satisfiability for data types in such a class. In particular, there are interesting parallels between the work on unification and satisfiability for lists, compact lists, sets, and HF sets in [35,34] and that in Section 7. Again, an important difference is that in [35,34] specific, inference-rule-based, unification and satisfiability algorithms are developed for each such data type, whereas in Section 7 both unification and satisfiability are obtainable as part of theory-generic, variant-based unification and satisfiability procedures. A detailed comparison between the two approaches should be a topic for further research.

There are also various results about decidability of QF or sometimes general first-order formulas in some initial unsorted, many-sorted, and order-sorted algebras modulo some equations, e.g., [69,29,30,9,31,77], that can be very useful, because, as shown in Section 6, they can be used in the reduction from satisfiability in an FVP initial algebra $T_{\Sigma/E}$ to satisfiability in $T_{\Omega/B_\Omega}$ by ensuring that satisfiability in $T_{\Omega/B_\Omega}$ is decidable. For example, as already mentioned, Theorem 8 generalizes to the OS and $ACCU$ case a similar result in [30] for the unsorted and $AC$ case for theories of constructors modulo axioms.

A line of work that is quite close in aims to the present one is the so-called *rewriting-based approach* to satisfiability [67,6,65,68,18,4,37]. Since the present work is also "rewriting-based" in an obvious sense, but quite different from the work just cited, to help the reader appreciate the differences I would rather

the sorts as a single universe sort $U$, which can be expressed as a surjective map of signatures $u : \Sigma \to \Sigma^u$, and at the formula level $\varphi \mapsto \varphi^u$ is the map that leaves all symbols unchanged except for changing the sort $s$ of each variable to the universe sort $U$. $u^{-1}$ is then the inverse relation associated to the formula map $u$; it ensures equi-satisfiability across the classes of order-sorted and unsorted models of the corresponding theories. The word "descent" should here be taken with large amounts of salt: *technically* we "descend" from $(\Sigma^u, \varnothing)$ to $(\Sigma, \varnothing)$; but *pragmatically* we really descend from (reduce the problem from) the more complicated $(\Sigma, \varnothing)$ to the simpler $(\Sigma^u, \varnothing)$, were standard congruence closure (resp. congruence closure modulo $AC$) can be used to solve the corresponding satisfiability problems. The point is that, unlike signature inclusions $\Sigma \subseteq \Sigma'$, $u : \Sigma \to \Sigma^u$ is *not* injective, and is actually surjective. Thus, the theory interpretation $u$, instead of moving us into a *richer* world as theory inclusions do, achieves a drastic reduction to a *simpler*, unsorted world.

call that work *superposition-based* satisfiability. That, is, the relevant first-order theory is axiomatized, and then it is proved that a superposition theorem proving inference system terminates for that theory together with any given set of ground clauses representing a satisfiability problem. Common features between the superposition-based and variant-based (both rewriting based!) approaches involve good modularity properties (see [4]), and no need for an explicit NO combination between procedures developed in either approach (although *both* approaches can of course be combined with other satisfiability procedures in the classical NO way[23]). The aims in both approaches are quite similar, but the methods are very different. I view both approaches as complementary and think that exploring potential synergies between them can further increase the extensibility of SMT solving.

Another approach to making SMT solving more extensible is presented in [36]. The goal is to allow a user to define a new theory with decidable QF satisfiability by axiomatizing it according to some requirements, and then making an SMT solver extensible by such a user-defined theory. This is done as follows:

1. A new theory $T'$, extending a given *background theory* $T$ already supported by the SMT solver, is axiomatized by the user in a first-order logic enhanced with the notion of using a literal $l$ as a *trigger* (or dually as a *witness*) in a formula $\varphi$, denoted $[l]\varphi$ (resp. $\langle l \rangle \varphi$).
2. If the user proves that $T'$ is *complete* and *terminating* in the precise sense of [36], he/she automatically obtains a QF satisfiability procedure for $T'$.
3. The DPLL(T) procedure is extended to support theories axiomatized by formulas with triggers. Thus, the satisfiability of a complete and terminating user-defined theory $T'$ can be decided. This extension of DPLL(T) has been implemented in the Alt-Ergo SMT solver [17], and a non-trivial case study on the decidable satisfiability of a theory of doubly-linked lists axiomatized with triggers using this implementation is presented in [36].

The approach in [36] is very different, yet complementary, to the one presented here. Ways of using both approaches together are worth investigating.

I have recently become aware of a technical report by Kapur and Zarba [64] on their so-called *reduction approach* to decision procedures that is closely related to the work in Section 8. Roughly speaking, the descent maps of Section 8 essentially correspond to the reduction functions of [64]. The two approaches can be brought closer together if the more general notion of descent function/relation sketched out in Footnotes 21–22 is considered, since then both approaches use theories $((\Sigma, \Pi), \mathcal{C})$ à la Ganzinger [49]. Although the methods in [64] and in Section 8 are quite similar, the specific results in each work are different: in [64] they show how the theories of lists, arrays, sets and bags can be reduced to those of Presburger arithmetic, constructors, and equality, and how disjoint reduction functions can be combined. Instead, in Section 8 the results focus around the notions of descent map extension and of proper theory extension,

[23] For combining variant-based decision procedures with other decision procedures, the *order-sorted* NO combination method in [90] will be particulary useful.

and use them to seamlessly combine natural and integer Presburger arithmetic with proper extensions. Furthermore, a variety of descent maps for arithmetic and other FVP theories are developed. A detailed exploration of how the ideas in [64] and in Section 8 can be combined and generalized using category theory seems very much worthwhile.

Last, but not least, there is also an important connection between the present work and a body of work in *inductive theorem proving* aimed at characterizing classes of algebraic specifications and associated kinds of formulas for which validity in an initial algebra can be decided automatically, e.g., [50,51,44,3]. The obvious relation to that work is that decidable validity and decidable satisfiability in an initial algebra are two sides of the same coin, so this paper might as well have been entitled "variant-based validity in initial algebras." What this work contributes to inductive theorem proving are new methods and results, complementing those in [50,51,44,3], for bringing large classes of initial algebras within the fold of decidable validity. In particular, to the best of my knowledge, the methods for getting decidable inductive validity in initial algebras in Sections 5–6, and for getting decidable inductive validity in *parameterized* data types presented in Section 7 seem to be new.

## 10 Conclusions and Future Work

This work has made three main contributions:

1. **To Unification Theory**: The new notion of *constructor unifier* can make the use of the generic variant-based unification algorithm considerably more efficient by generating fewer unifiers than up to now. This can have a substantial impact in reducing the search space of variant-unification-based model checking methods such as those used in, e.g., [41,11].
2. **To Extensible Satisfiability Methods**: The new *theory-generic algorithm for variant-based satisfiability* presented in this paper brings an infinite class of theories for which satisfiability in their initial algebras is decidable within the fold of SMT solving, thus making SMT solving considerably more extensible. Such theories are in fact *user-definable*, their required properties easy to check (by existing methods and tools for checking confluence, termination, sufficient completeness, and FVP), and quite modular. Specifically, the classes of theories to which these methods can be applied to make satisfiability in their initial algebras decidable has been extended in four concentric circles: (i) theories $(\Omega, ACCU)$, which are all OS-compact; (ii) FVP theories having a constructor decomposition of type (i); (iii) parameterized data types (several examples have been given to illustrate the general method) that transform input theories with an OS-compact core into corresponding instantiations of the parameterized data type, also having an OS-compact core, including input theories such as those in (ii), and *nested* instantiations of different parameterized data types; and (iv) a still broader class of theories that can be reduced to cases (i)–(iii) by means of *descent maps*.

3. **To Relating Satisfiability Across Theories**: The notions of *descent map*, *descent map extension*, and *proper theory extension* make it easy to : (i) relate satisfiability across different theories, reducing the satisfiability checking problem in a more complex theory to that in simpler, already known one; (ii) specify satisfiability algorithms in a modular way as *compositions* of several simpler descent maps; and (iii) increasing the range and efficiency of satisfiability algorithms by mapping their theories to corresponding core theories, which may have considerably more efficient satisfiability algorithms.

Besides extensibility, two other *key advantages* of variant satisfiability can be pointed out. The first one might be called its *seamless compositionality*, without any need for the seams and expense of NO combinations: if the initial models of $T$ and $T'$ have decidable variant-based QF satisfiability, that of their *union* $T \cup T'$ does too, under mild composed termination requirements. This is already a key advantage of variant-based unification, here extended to initial satisfiability. Furthermore, Theorems 14–15 have illustrated as concrete examples a general method by which descent map extensions can be used to provide seamless combinations for FVP theories lacking OS-compact constructor decompositions.

A second key advantage can be made explicit dialectically. Suppose you are type skeptical and at this point in the paper would like to ask: why did I have to put up with all this order-sorted stuff? Couldn't you just do all this in good old (untyped) first-order logic using good old (untyped) rewriting? Isn't all this some kind of extravagant fixation with types? I would then answer you by pointing out a few things: (i) the unsorted case is a *special subcase*, so *a priori* you didn't lose anything, since everything specializes to your case; (ii) the extra cost of carrying types around is fairly modest: are you not just type skeptical but also *generality impaired*? (iii) you can *insist* in your rejection of types, but you will have to live with the consequences of type poverty, which in this case are quite substantial. A key method to get decidable satisfiability has been to specify data types having algebras of constructors that are *free* modulo $ACCU$. But this is just *impossible* to get for many examples, not just in an untyped setting, but even in a simply typed (many-sorted) one. Specifically, you would automatically lose Examples 10, 13–15 and 17, and would have a harder time getting decidable satisfiability for many other examples. In short, besides the obvious greater expressiveness and flexibility, a key advantage of an order-sorted setting is that it makes it much easier to prove compactness of the constructor initial algebras.

Much work remains ahead. I have already pointed out that variant-based satisfiability *complements*, and can be synergistic with, other methods, such as superposition-based satisfiability, decidable theories defined by means of formulas with triggers, or the NO combination method. Indeed, NO combinations remain essential, since one obviously wants to combine generic procedures based on variant-based, superpositon-based, or trigger-based algorithms, with efficiently implemented ones for well-known theories and with each other. In this regard, my focus in this work on satisfiability in *initial algebras* could be misunderstood as exclusive, when actually it is not. The general picture emerging from such NO combinations is that of combinations of theories which may have some "ini-

tiality constraints" (more generally understood as *freeness* constraints, as in the case of formulas valid in *uninstantiated* parameterized data types, which I have mentioned *en passant* in various examples of parameterized data types) as well as some other unconstrained theories with a "loose semantics," in the sense of Goguen and Burstall [52], such as the theory of uninterpreted function symbols.

This suggests the longer-term goal of developing an *extensible framework and tools* for the definition, prototyping and combination of satisfiability procedures. Within such a framework one would already have available a library of dedicated and generic procedures that would make it quite easy for *users* to *prototype* a first version of a new satisfiability procedure by *combining* existing procedures with a newly specified one. There are of course tensions and tradeoffs between the efficiency of a generic algorithm and that of an optimized, domain-specific one; but the whole point of an extensible framework is precisely to make it easy to migrate in a *correct*, tool-supported, and seamless way prototypes into efficient algorithms. In this regard, *descent maps* can provide a useful migration method that can be applied very broadly to both generic and dedicated algorithms, and to quantified and unquantified formulas. Also, the computational cost of deciding satisfiability is seldom that of a single procedure but is instead the *overall* cost. Here interesting situations may arise. For example, we may have a combination of five procedures obtained by theory-generic methods and two by dedicated algorithms. Although the dedicated ones will typically be more efficient, since the five generic ones may be combined as their *union*, NO will only have to deal with the interactions between *three* procedures, as opposed to seven, thus reducing the computational cost of the combination.

On a shorter time frame, the algorithms presented here, and suitable extensions or optimizations of them, should be implemented; and new descent maps should be developed. Work on new descent maps still remains in the future, but a more detailed design of the variant satisfiability procedure and its key auxiliary algorithms, including correctness proofs, as well as a first prototype in Maude, have recently been developed [85,86]. Current work involves both experimentation with, and improvements of, this prototype. As a more mature implementation is reached, it will become possible to carry out performance experiments and to compare variant-based satisfiability with other existing methods and tools such as those for superposition-based and trigger-based satisfiability [67,6,65,68,18,4,37,36], constraint logic programming methods such as those in [35,34] and others, and state of the art SMT solvers.

Last, but not least, besides experimental performance comparisons, *computational complexity bounds* should be developed whenever possible. A specific complexity measure is of course impossible for theory-generic algorithm such as variant-based narrowing, unification, and satisfiability, just as for other theory-generic algorithms or semi-algorithms like superposition theorem proving, or trigger-based satisfiability algorithms, because in all these cases the complexity depends on the input theory. However, complexity measures can sometimes be possible when the input theory $T$ is fixed. For example, in superposition theorem proving, results along the lines of [15,67,6,4] do exactly this.

For variant-based satisfiability this will be a highly non-trivial task, because — besides the fact that complexity issues for variant-based computations have not yet been investigated— all $R, B$-variant-based computations first of all invoke order-sorted $B$-unification algorithms. But these $B$-unification algorithms are themselves *combinations* of simpler ones for theories such as free symbols, $C$, $AC$, $ACU$, $U$, and so on, so that their complexity depends not just on that of the component algorithms, but also on that of the, quite involved, combination infrastructure. To further complicate things, we cannot just use the complexity of their unsorted unification versions: the added complexity of their sort computations must be included, which itself depends on the given subsort hierarchy (see [39] for a detailed complexity analysis of order-sorted unification when only free function symbols are involved).

**Acknowledgements**. I thank the organizers of FTSCS 2015 for inviting me to present these ideas in Paris, and the FTSCS participants for their interest and very helpful comments. I thank Andrew Cholewa, Steven Eker, Santiago Escobar, Ralf Sasse, and Carolyn Talcott for their contributions to the development of the theory and Maude implementation of folding variant narrowing. I have learned much about satisfiability from Maria-Paola Bonacina, Vijay Ganesh and Cesare Tinelli along many conversations; I am most grateful to them for their kind enlightenment. I also thank the following persons for their very helpful comments on earlier drafts: Maria-Paola Bonacina, Santiago Escobar, Dorel Lucau, Peter Ölveczky, Vlad Rusu, Ralf Sasse, Natarajan Shankar, and Cesare Tinelli. The pioneering work of Hubert Comon-Lundh about compact theories [30], and that of him with Stephanie Delaune about the finite variant property [32], have both been important sources of inspiration for the ideas presented here. This work has been partially supported by NSF Grant CNS 13-19109.

## References

1. Alpuente, M., Escobar, S., Iborra, J.: Termination of narrowing revisited. Theor. Comput. Sci. 410(46), 4608–4625 (2009)
2. Alpuente, M., Escobar, S., Iborra, J.: Modular termination of basic narrowing and equational unification. Logic Journal of the IGPL 19(6), 731–762 (2011)
3. Aoto, T., Stratulat, S.: Decision procedures for proving inductive theorems without induction. In: Proc. PPDP2014. pp. 237–248. ACM (2014)
4. Armando, A., Bonacina, M.P., Ranise, S., Schulz, S.: New results on rewrite-based satisfiability procedures. ACM Trans. Comput. Log. 10(1) (2009)
5. Armando, A., Castellini, C., Giunchiglia, E.: SAT-based procedures for temporal reasoning. In: Biundo, S., Fox, M. (eds.) Proceedings of the 5th European Conference on Planning (Durham, UK). Lecture Notes in Computer Science, vol. 1809, pp. 97–108. Springer (2000)
6. Armando, A., Ranise, S., Rusinowitch, M.: A rewriting approach to satisfiability procedures. Inf. Comput. 183(2), 140–164 (2003)
7. Audemard, G., Bertoli, P., Cimatti, A., Korniłowicz, A., Sebastiani, R.: A SAT-based approach for solving formulas over boolean and linear mathematical propositions. In: Voronkov, A. (ed.) Proceedings of the 18th International Conference

on Automated Deduction. Lecture Notes in Artificial Intelligence, vol. 2392, pp. 195–210. Springer (2002)
8. Baader, F., Schulz, K.: Unification in the union of disjoint equational theories: combining decision procedures. Journal of Symbolic Computation 21, 211–243 (1996)
9. Baader, F., Schulz, K.U.: Combination techniques and decision problems for disunification. Theor. Comput. Sci. 142(2), 229–255 (1995)
10. Baader, F., Schulz, K.U.: Combining constraint solving. In: Constraints in Computational Logics CCL'99, International Summer School. vol. 2002, pp. 104–158. Springer LNCS (1999)
11. Bae, K., Meseguer, J.: Infinite-state model checking of LTLR formulas using narrowing. In: Rewriting Logic and Its Applications - 10th International Workshop, WRLA 2014, Held as a Satellite Event of ETAPS, Grenoble, France, April 5-6, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8663, pp. 113–129. Springer (2014)
12. Barrett, C., Shikanian, I., Tinelli, C.: An abstract decision procedure for satisfiability in the theory of inductive data types. Journal on Satisfiability, Boolean Modeling and Computation 3, 21–46 (2007)
13. Barrett, C., Tinelli, C.: Satisfiability modulo theories. In: Clarke, E., Henzinger, T., Veith, H. (eds.) Handbook of Model Checking. Springer (2014), (to appear)
14. Barrett, C.W., Dill, D.L., Stump, A.: Checking satisfiability of first-order formulas by incremental translation to SAT. In: Godskesen, J.C. (ed.) Proceedings of the International Conference on Computer-Aided Verification. Lecture Notes in Computer Science (2002)
15. Basin, D.A., Ganzinger, H.: Automated complexity analysis based on ordered resolution. J. ACM 48(1), 70–109 (2001)
16. Bloom, S., Tindell, R.: Varieties of if-then-else. SIAM Journal of Computing 12, 677–707 (1983)
17. Bobot, F., Conchon, S., Contejean, E., Lescuyer, S.: Implementing polymorphism in smt solvers. In: Proc. 6th Intl. Workshop on Satisfiability Modulo Theories and 1st International Workshop on Bit-Precise Reasoning. pp. 1–5. SMT '08/BPR '08, ACM (2008)
18. Bonacina, M.P., Echenim, M.: On variable-inactivity and polynomial $\mathcal{T}$-satisfiability procedures. J. Log. Comput. 18(1), 77–96 (2008)
19. Bouchard, C., Gero, K.A., Lynch, C., Narendran, P.: On forward closure and the finite variant property. In: Proc. FroCoS 2013. LNCS, vol. 8152, pp. 327–342. Springer (2013)
20. Boudet, A.: Combining unification algorithms. J. Symb. Comput. 16(6), 597–626 (1993)
21. Bradley, A.R., Manna, Z.: The calculus of computation - decision procedures with applications to verification. Springer (2007)
22. Bryant, R.E., Lahiri, S.K., Seshia, S.A.: Modeling and verifying systems using a logic of counter arithmetic with lambda expressions and uninterpreted functions. In: Computer Aided Verification, 14th International Conference, CAV 2002,Copenhagen, Denmark, July 27-31, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2404, pp. 78–92. Springer (2002)
23. Chadha, R., Ştefan Ciobâcă, Kremer, S.: Automated verification of equivalence properties of cryptographic protocols. In: Proc. ESOP 2012. vol. 7211, pp. 108–127. Springer LNCS (2012)
24. Cholewa, A., Meseguer, J., Escobar, S.: Variants of variants and the finite variant property. Tech. rep., CS Dept. University of Illinois at Urbana-Champaign (February 2014), available at `http://hdl.handle.net/2142/47117`

25. Ciobaca., S.: Verification of Composition of Security Protocols with Applications to Electronic Voting. Ph.D. thesis, ENS Cachan (2011)
26. Clavel, M., Durán, F., Eker, S., Meseguer, J., Lincoln, P., Martí-Oliet, N., Talcott, C.: All About Maude – A High-Performance Logical Framework. Springer LNCS Vol. 4350 (2007)
27. Cohen, P.: Set Theory and the Continuum Hypothesis. W.A. Benjamin (1966)
28. Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata` (2007), Release October, 12th 2007
29. Comon, H., Lescanne, P.: Equational problems and disunification. Journal of Symbolic Computation 7, 371–425 (1989)
30. Comon, H.: Complete axiomatizations of some quotient term algebras. Theor. Comput. Sci. 118(2), 167–191 (1993)
31. Comon, H., Delor, C.: Equational formulae with membership constraints. Inf. Comput. 112(2), 167–216 (1994)
32. Comon-Lundth, H., Delaune, S.: The finite variant property: how to get rid of some algebraic properties, in Proc *RTA'05*, Springer LNCS 3467, 294–307, 2005
33. Dershowitz, N., Jouannaud, J.P.: Rewrite systems. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, Vol. B, pp. 243–320. North-Holland (1990)
34. Dovier, A., Piazza, C., Rossi, G.: A uniform approach to constraint-solving for lists, multisets, compact lists, and sets. ACM Trans. Comput. Log. 9(3) (2008)
35. Dovier, A., Policriti, A., Rossi, G.: A uniform axiomatic view of lists, multisets, and sets, and the relevant unification algorithms. Fundam. Inform. 36(2-3), 201–234 (1998)
36. Dross, C., Conchon, S., Kanig, J., Paskevich, A.: Adding Decision Procedures to SMT Solvers using Axioms with Triggers. Journal of Automated Reasoning (2016), `https://hal.archives-ouvertes.fr/hal-01221066`, accepted for publication
37. Echenim, M., Peltier, N.: An instantiation scheme for satisfiability modulo theories. J. Autom. Reasoning 48(3), 293–362 (2012)
38. Ehrig, H., Mahr, B.: Fundamentals of Algebraic Specification 1. Springer (1985)
39. Eker, S.: Fast sort computations for order-sorted matching and unification. In: Formal Modeling: Actors, Open Systems, Biological Systems - Essays Dedicated to Carolyn Talcott on the Occasion of Her 70th Birthday. vol. 7000, pp. 299–314. Springer LNCS (2011)
40. Erbatur, S., Escobar, S., Kapur, D., Liu, Z., Lynch, C., Meadows, C., Meseguer, J., Narendran, P., Santiago, S., Sasse, R.: Asymmetric unification: A new unification paradigm for cryptographic protocol analysis. In: Bonacina, M.P. (ed.) CADE. Lecture Notes in Computer Science, vol. 7898, pp. 231–248. Springer (2013)
41. Escobar, S., Meadows, C., Meseguer, J.: Maude-NPA: cryptographic protocol analysis modulo equational properties. In: Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009 Tutorial Lectures, LNCS, vol. 5705, pp. 1–50. Springer (2009)
42. Escobar, S., Sasse, R., Meseguer, J.: Folding variant narrowing and optimal variant termination. In: Proc. WRLA 2010. vol. 6381, pp. 52–68 (2010), springer LNCS
43. Escobar, S., Sasse, R., Meseguer, J.: Folding variant narrowing and optimal variant termination. J. Algebraic and Logic Programming 81, 898–928 (2012)
44. Falke, S., Kapur, D.: Rewriting induction + linear arithmetic = decision procedure. In: Proc. IJCAR 2012. vol. 7364, pp. 241–255. Springer LNCS (2012)
45. Fay, M.: First-order unification in an equational theory. In: Proceedings of the 4th Workshop on Automated Deduction. pp. 161–167 (1979)

46. Filliâtre, J.C., Owre, S., Rueß, H., Shankar, N.: Ics: Integrated canonizer and solver. In: Berry, G., Comon, H., Finkel, A. (eds.) Proceedings of the 13th International Conference on Computer Aided Verification (Paris, France). Lecture Notes in Computer Science, vol. 2102, pp. 246–249. Springer-Verlag (July 2001)
47. Flanagan, C., Joshi, R., Ou, X., Saxe, J.B.: Theorem proving using lazy proof explication. In: Jr., W.A.H., Somenzi, F. (eds.) Proceedings of the 15th International Conference on Computer Aided Verification. Lecture Notes in Computer Science, vol. 2725, pp. 355–367. Springer (2003)
48. Gallier, J.H., Snyder, W.: Complete sets of transformations for general E-unification. Theor. Comput. Sci. 67(2&3), 203–260 (1989)
49. Ganzinger, H.: Shostak light. In: Proc. CADE 2002. vol. 2392, pp. 332–346. Springer LNCS (2002)
50. Giesl, J., Kapur, D.: Decidable classes of inductive theorems. In: Proc. IJCAR 2001. vol. 2083, pp. 469–484. Springer LNCS (2001)
51. Giesl, J., Kapur, D.: Deciding inductive validity of equations. In: Proc. CADE 2003. vol. 2741, pp. 17–31. Springer LNCS (2003)
52. Goguen, J., Burstall, R.: Institutions: Abstract model theory for specification and programming. Journal of the ACM 39(1), 95–146 (1992)
53. Goguen, J., Meseguer, J.: Models and equality for logical programming. In: Ehrig, H., Levi, G., Kowalski, R., Montanari, U. (eds.) Proceedings TAPSOFT'87, Springer LNCS, vol. 250, pp. 1–22. Springer-Verlag (1987)
54. Goguen, J., Meseguer, J.: Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. Theoretical Computer Science 105, 217–273 (1992)
55. González-Burgueño, A., Santiago, S., Escobar, S., Meadows, C., Meseguer, J.: Analysis of the IBM CCA security API protocols in maude-npa. In: Proc. SSR 2014. vol. 8893, pp. 111–130. Springer LNCS (2014)
56. Gramlich, B.: Modularity in term rewriting revisited. Theor. Comput. Sci. 464, 3–19 (2012)
57. Hendrix, J., Clavel, M., Meseguer, J.: A sufficient completeness reasoning tool for partial specifications. In: Rewriting Techniques and Applications, $16^{th}$ Intl. Conference RTA 2005. vol. 3467, pp. 165–174. Springer LNCS (2005)
58. Hendrix, J., Meseguer, J.: Order-sorted equational unification revisited. Electr. Notes Theor. Comput. Sci. 290, 37–50 (2012)
59. Hendrix, J., Meseguer, J., Ohsaki, H.: A sufficient completeness checker for linear order-sorted specifications modulo axioms. In: Automated Reasoning, Third International Joint Conference, IJCAR 2006. pp. 151–155 (2006)
60. Hullot, J.M.: Canonical forms and unification. In: Bibel, W., Kowalski, R. (eds.) Proceedings, Fifth Conference on Automated Deduction, pp. 318–334. Springer-Verlag (1980), lNCS, Volume 87
61. Jouannaud, J.P., Kirchner, C., Kirchner, H.: Incremental construction of unification algorithms in equational theories. In: Proc. ICALP'83. pp. 361–373. Springer LNCS 154 (1983)
62. Jouannaud, J.P., Kirchner, H.: Completion of a set of rules modulo a set of equations. SIAM Journal of Computing 15, 1155–1194 (November 1986)
63. Kapur, D., Narendran, P.: Complexity of unification problems with associative-commutative operators. J. Autom. Reasoning 9(2), 261–288 (1992)
64. Kapur, D., Zarba, C.G.: A reduction approach to decision procedures (2005), `http://www.cs.unm.edu/ kapur/mypapers/reduction.pdf`

65. Kirchner, H., Ranise, S., Ringeissen, C., Tran, D.: On superposition-based satisfiability procedures and their combination. In: Proc. ICTAC 2005. vol. 3722, pp. 594–608. Springer LNCS (2005)
66. Krstic, S., Goel, A., Grundy, J., Tinelli, C.: Combined satisfiability modulo parametric theories. In: Proc. TACAS 2007. vol. 4424, pp. 602–617. Springer LNCS (2007)
67. Lynch, C., Morawska, B.: Automatic decidability. In: Proc. LICS 2002. p. 7. IEEE Computer Society (2002)
68. Lynch, C., Tran, D.: Automatic decidability and combinability revisited. In: Proc. CADE 2007. vol. 4603, pp. 328–344. Springer LNCS (2007)
69. Maher, M.J.: Complete axiomatizations of the algebras of finite, rational and infinite trees. In: Proc. LICS '88. pp. 348–357. IEEE Computer Society (1988)
70. Meseguer, J.: Order-sorted parameterization and induction. In: Semantics and Algebraic Specification. Lecture Notes in Computer Science, vol. 5700, pp. 43–80. Springer (2009)
71. Meseguer, J.: Strict coherence of conditional rewriting modulo axioms. Tech. Rep. `http://hdl.handle.net/2142/50288`, C.S. Department, University of Illinois at Urbana-Champaign (August 2014)
72. Meseguer, J.: Variant-based satisfiability in initial algebras. In: Artho, C., Ölveczky, P. (eds.) proc. FTSCS 2015. pp. 1–32. Springer CCIS 596 (2016)
73. Meseguer, J.: Membership algebra as a logical framework for equational specification. In: Proc. WADT'97. pp. 18–61. Springer LNCS 1376 (1998)
74. Meseguer, J.: Order-sorted rewriting and congruence closure. In: Proc. FOSSACS 2016. Lecture Notes in Computer Science, vol. 9634, pp. 493–509. Springer (2016)
75. Meseguer, J., Goguen, J.: Order-sorted algebra solves the constructor-selector, multiple representation and coercion problems. Information and Computation 103(1), 114–158 (1993)
76. Meseguer, J., Guessarian, I.: On the axiomatization of if-then-else. SIAM Journal of Computing 16, 332–357 (1987)
77. Meseguer, J., Skeirik, S.: Equational formulas and pattern operations in initial order-sorted algebras. In: Falaschi, M. (ed.) Proc. LOPSTR 2015. vol. 9527, pp. 36–53. Springer LNCS (2015)
78. de Moura, L., Rueß, H.: Lemmas on demand for satisfiability solvers. In: Proc. of the Fifth International Symposium on the Theory and Applications of Satisfiability Testing (SAT'02) (May 2002)
79. Nelson, G., Oppen, D.C.: Simplification by cooperating decision procedures. ACM Trans. Program. Lang. Syst. 1(2), 245–257 (1979)
80. Nelson, G., Oppen, D.C.: Fast decision procedures based on congruence closure. J. ACM 27(2), 356–364 (Apr 1980)
81. Nieuwenhuis, R., Oliveras, A., Tinelli, C.: Solving SAT and SAT Modulo Theories: from an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T). Journal of the ACM 53(6), 937–977 (Nov 2006)
82. Oppen, D.C.: Complexity, convexity and combinations of theories. Theor. Comput. Sci. 12, 291–302 (1980)
83. Schmidt, B., Meier, S., Cremers, C.J.F., Basin, D.A.: Automated analysis of Diffie-Hellman protocols and advanced security properties. In: Proc. CSF 2012. pp. 78–94. IEEE (2012)
84. Shostak, R.E.: Deciding combinations of theories. Journal of the ACM 31(1), 1–12 (Jan 1984)

85. Skeirik, S., Meseguer, J.: Metalevel algorithms for variant-based satisfiability. In: Lucanu, D. (ed.) Proc. WRLA 2016. Springer LNCS (2016), to appear. Preliminary version in `https://fmse.info.uaic.ro/events/WRLA2016/`
86. Skeirik, S., Meseguer, J.: Metalevel algorithms for variant-based satisfiability. Tech. Rep. `http://hdl.handle.net/2142/90238`, University of Illinois at Urbana-Champaign (June 2016)
87. Slagle, J.R.: Automated theorem-proving for theories with simplifiers commutativity, and associativity. J. ACM 21(4), 622–642 (1974)
88. Snyder, W.: A Proof Theory for General Unification. Birkhäuser (1991)
89. Stump, A., Barrett, C.W., Dill, D.L., Levitt, J.R.: A decision procedure for an extensional theory of arrays. In: Proc. LICS 2001. pp. 29–37. IEEE Computer Society (2001)
90. Tinelli, C., Zarba, C.G.: Combining decision procedures for sorted theories. In: Proc. JELIA 2004. vol. 3229, pp. 641–653. Springer LNCS (2004)
91. Toyama, Y.: Counterexamples to termination for the direct sum of term rewriting systems. Inf. Process. Lett. 25(3), 141–143 (1987)
92. Yang, F., Escobar, S., Meadows, C., Meseguer, J., Narendran, P.: Theories of homomorphic encryption, unification, and the finite variant property. In: Proc. PPDP 2014. pp. 123–133. ACM (2014)

## A Proof of Proposition 1

*Proof.* First of all observe that —since for any order-sorted signature, under very general assumptions on axioms $B$ which are satisfied by those above for $ACCU$— any order-sorted $ACCU$-unifier is a variable specialization of some unsorted $ACCU$-unifier, [58], and, by $(S, \leqslant)$ locally finite, there is, up to variable renaming, a finite number of specializations for each variable, we can, without loss of generality, prove the result for the special case when $\Omega$ is unsorted. Furthermore, since the theories $AC$, $C$, $LU$, $RU$, $U$, $CU$ and $ACU$ (and of course free function symbols), have all finitary unification algorithms, we can decompose the given $ACCU$ axioms $B$ into a disjoint union $B = \biguplus_f B_f$, where $B_f$ denotes the axioms for symbol $f$, and applying unification combination methods such as those in [8], ensure *a priori* that the set of most complete unsorted $ACCU$-unifiers $Unif_{ACCU}(u = v)$ for *any* equation $u = v$ with any number of variables if finite. And, by the above remarks, the same holds in the order-sorted case. Therefore, the real issue to be shown is that when $u = v$ is $ACCU$-non-trivial and only one variable $x$ appears in the disequation, the most general unifiers in $Unif_{ACCU}(u = v)$ are all *ground*. However, I will prove both finiteness and groundedness anyway, without appealing to the above-mentioned unification combination methods.

We reason by induction on $n = max(|u|, |v|)$, with $|t|$ the size of $t$ as a tree. To simplify life I make no distinction between $u = v$ and $v = u$. If $n = 1$, since $u = v$ is $ACCU$-non-trivial, we must have an equation of the form $x = a$, with $a$ a constant, which has $\{x \mapsto a\}$ as the only possible $ACCU$-unifier. Assuming the result for $1 \leqslant n' \leqslant n$, let us prove it for $n + 1 = max(|u|, |v|)$, so that $n + 1 \geqslant 2$. We reason by cases. First we consider the cases when the axioms holding for $f$ and $g$ in either an equation $x = f(u_1, \ldots, u_n)$, or an equation

$f(u_1, \ldots, u_n = g(v_1, \ldots, v_m)$ are in $ACC = ACCU - U$, that is, no $U$, or $LU$, or $RU$ axioms hold for either $f$ or $g$. Then we consider the cases where some $U$, or $LU$, or $RU$ axiom holds for $f$, or $g$, or both:

1. $x = f(u_1, \ldots, u_n)$, and then: (i) if $x$ occurs in $f(u_1, \ldots, u_n)$ there is no $ACCU$-unifier, because the $ACC$ axioms holding for $f$ are size-preserving, so any minimal-size solution for $x$ in the $ACCU$-equivalence class would then also have an even *smaller* size; and (ii) if $f(u_1, \ldots, u_n)$ is a ground term, the only possible $ACCU$-unifier up to $ACCU$-equivalence ($ACC$-equivalence classes are finite) is $\{x \mapsto f(u_1, \ldots, u_n)\}$.
2. In $f(u_1, \ldots, u_n = g(v_1, \ldots, v_m)$, with the axioms of $f$ and $g$ in $ACC$ and $f \neq g$ (or $n \neq m$), since only such axioms can be applied at the top of each term and they will never change the top function symbol, there is no solution, so we can reduce to the case $f(u_1, \ldots, u_k) = f(v_1, \ldots, u_k)$. Let us first deal with the case where $f$ is a free function symbol. Then the $ACCU$-unifiers of $f(u_1, \ldots, u_k) = f(v_1, \ldots, v_k)$ are exactly those of $u_1 = v_1 \wedge \ldots \wedge u_k = v_k$, and since $f(u_1, \ldots, u_k) \neq_{ACCU} f(v_1, \ldots, v_k)$, we must have some $u_i \neq_{ACCU} v_i$, where if both terms are gound, there is no $ACCU$-unifier, and otherwise by the induction hypothesis $u_i \neq_{ACCU} v_i$ has a finite number of ground $ACC$-unifiers, which must contain the $ACCU$-unifiers of $f(u_1, \ldots, u_k) = f(v_1, \ldots, u_k)$ as a subset.
3. If $f$ is commutative, the equation is of the form $f(u_1, u_2) = f(v_1, v_2)$ and its $ACCU$-unifiers are exactly those of $(u_1 = v_1 \wedge u_2 = v_2) \vee (u_1 = v_2 \wedge u_2 = v_1)$. Thus, reasoning as in (2) above and applying the induction hypothesis $f(u_1, u_2) = f(v_1, v_2)$ has a finite number of ground $ACCU$-unifiers.
4. If $f = +$ is associative-commutative, we can represent $u$ and $v$ in flattened form (as unparenthesized additions of two or more "alien subterms") and have two cases:
   (a) $u = u_1 + \ldots + u_k$ and $v = v_1 + \ldots + v_{k'}$, where none of the $u_i, v_j$ is the variable $x$, and their top function symbols are all different from $+$. Then if $k \neq k'$ the equation has no $ACCU$-unifier, and otherwise the $ACCU$-unifiers of $u = v$ are exactly those of $\bigvee_{\sigma \in Perm(k)} \bigwedge_{1 \leqslant i \leqslant k} u_i = v_{\sigma(i)}$, where $Perm(k)$ denotes the set of permutations of $k$ elements. Again, since $u \neq_{ACCU} v$, for each $\sigma$ we must have an index $i$ such that $u_i \neq_{ACCU} v_{\sigma(i)}$. If $u_i$ and $v_{\sigma(i)}$ are ground terms, there is no $ACCU$-unifier for $\bigwedge_{1 \leqslant i \leqslant k} u_i = v_{\sigma(i)}$; otherwise, the induction hypothesis applies and the $ACCU$-unifiers of $u_i =_{ACC} v_{\sigma(i)}$ are ground $ACCU$-unifiers; and the $ACCU$-unifiers of $\bigwedge_{1 \leqslant i \leqslant k} u_i = v_{\sigma(i)}$ are a subset of them. Therefore, all $ACCU$-unifiers of $\bigwedge_{1 \leqslant i \leqslant k} u_i = v_{\sigma(i)}$ are ground, so that all $ACCU$-unifiers of $u = v$ are also ground.
   (b) $u = m \cdot x + u_1 + \ldots + u_k$ and $v = m' \cdot x + v_1 + \ldots + v_{k'}$, where $m + m' \geqslant 1$, and $m \cdot x$ abbreviates $x + \overset{m}{\ldots} + x$, for $x$ a variable, none of the $u_i, v_j$ is the variable $x$, and their top function symbols are all different from $+$. If $m = m'$ then, this equation is $ACCU$-unifiable iff $u = v$ is so, but then, (i) if $u$ and $v$ are ground terms, there is no $ACCU$-unifier; and otherwise both equations have the same $ACCU$-unifiers, which are exactly those

of $u_1 + \ldots + u_k = v_1 + \ldots + v_{k'}$, which is case (a) already taken care of. Otherwise, assume without loss of generality $m > m'$, and then the $ACCU$-unifiers of $u = v$ are exactly those of $(m-m')\cdot x + u_1 + \ldots + u_k = v_1 + \ldots + v_{k'}$. But if $(m-m')\cdot x + u_1 + \ldots + u_k = v_1 + \ldots + v_{k'}$ has an $ACCU$-unifier, then there must be an $l$, $l \geqslant 1$, such that $((m-m')\cdot l) + k = k'$, and such an $ACCU$-unifier must, up to $ACC$-equivalence, be of the form $x \mapsto (v_{j_1} + \ldots + v_{j_l})$ with $1 \leqslant j_1 < \ldots < j_l \leqslant k'$ and with $v_{j_1} + \ldots + v_{j_l}$ ground. Since there is a finite number of such choices for the $v_{j_1}, \ldots, v_{j_l}$, there is also a finite number of possible ground $ACCU$-unifiers for $u = v$.

5. $x = f(u_1, u_2)$ with $f$ $ACCU$ and satisfying some $LU$, or $RU$ or $U$ axiom with unit element $e$. Since the most general situations arise in the $U$ case, I leave the more special $LU$, or $RU$ cases for the reader. Then: (i) if $x$ occurs in $f(u_1, u_2)$ for $x = f(u_1, u_2)$ to be $ACCU$-unifiable we must have either $u_1 =_{ACCU} x$ and $u_2 =_{ACCU} e$, or the other way around, or $u_1 =_{ACCU} u_2 =_{ACCU} x$; and then, up to $ACCU$-equivalence, the only unifier is $\{x \mapsto e\}$; and (ii) if $f(u_1, \ldots, u_n)$ is a ground term, the only possible $ACCU$-unifier up to $ACCU$-equivalence is $\{x \mapsto f(u_1, \ldots, u_n)\}$.
6. If the equation is of the form $f(u_1, u_2) = f(v_1, v_2)$ where $f$ only satisfies either the $LU$, or $RU$ or $U$ axioms, this equation will have the same $ACCU$-unifiers as one where we have applied all the $LU$, $RU$, or $U$ axioms for *any* $g$ in $\Sigma$ with unit $e_g$ (including $f$ itself with unit $e$) as rewrite rules (modulo $ACCU$) $g(e_g, x) \to x$, or $g(x, e_g) \to x$, or both and reducing to normal form (call it the *$U$-normal form* of a term). Since these rules are term-size-decreasing rules and $ACC$ axioms are term-size-preserving, a $U$-normal form will have the smallest size possible in its $ACCU$-equivalence class. Therefore, we may assume without loss of generality that none of the $u_1, u_2, v_1, v_2$ is $e$ or can be $ACCU$-equal to $e$, since otherwise an equation between terms respectively $ACCU$-equivalent to $f(u_1, u_2)$ and $f(v_1, v_2)$ but of *smaller* size exists and the induction hypothesis applies. But then this reduces the problem to the free function symbol case (2) above.
7. If the equation is of the form $f(u_1, u_2) = f(v_1, v_2)$ where $f$ is commutative, the additional $LU$, $RU$, or $U$ axiom cases generate the same equality relation as the $U$ case. By the same reasoning as in (6) we may assume without loss of generality that $f(u_1, u_2)$ and $f(v_1, v_2)$ are in $U$-normal form, so that none of the $u_1, u_2, v_1, v_2$ is $e$ or can be $e$ $ACCU$-equal to $e$. But then this reduces the problem to the commutative-only case (3) above.
8. If $f = +$ satifies the associative-commutative and identity element 0 axioms we can again assume withou loss of generality that both sides are in $U$-canonical form. We then have two cases:
   (a) $u = u_1 + \ldots + u_k$ and $v = v_1 + \ldots + v_{k'}$, where none of the $u_i, v_j$ is the variable $x$ or $ACCU$-equal to 0, and their top function symbols are all different from $+$. But then the problem is reduced to the $AC$-only case (4)-(a).
   (b) $u = m\cdot x + u_1 + \ldots + u_k$ and $v = m'\cdot x + v_1 + \ldots + v_{k'}$, where $m + m' \geqslant 1$, and none of the $u_i, v_j$ is the variable $x$ or $ACCU$-equal to 0, and their top function symbols are all different from $+$. If $m = m'$ then, this

equation is $ACCU$-unifiable iff $u = v$ is so, but then, (i) if $u$ and $v$ are ground terms, there is no $ACCU$-unifier; and otherwise both equations have the same $ACCU$-unifiers, which are exactly those of $u_1 + \ldots + u_k = v_1 + \ldots + v_{k'}$, which is case (8)-(a) already reduced to (4)-(a). Otherwise, assume without loss of generality $m > m'$, and then the $ACCU$-unifiers of $u = v$ are exactly those of $(m-m')\cdot x + u_1 + \ldots + u_k = v_1 + \ldots + v_{k'}$. Then we can distinguish two cases: (i) if $k = k'$, the only $ACCU$-unifier possible up to $ACCU$-equivalence exists only when all the $u_i, v_j$ are ground and $u_1 + \ldots + u_k =_{ACCU} v_1 + \ldots + v_k$, and is the unifier $\{x \mapsto 0\}$; and (ii) if $k \neq k'$ the problem then reduces to the exact same $AC$-subcase in (4)-(b).

This finishes the proof of the proposition. □

## B Finite Sorts in Initial Algebras Modulo *ACCU*

Consider an order-sorted theory $(\Omega, ACCU)$, and let the axioms $ACCU$ decompose as a disjoint union $ACCU = B \uplus U$, where $B$ are the $C$ or $AC$ axioms, and $U$ are the unit axioms. By the assumption that $\Omega$ is $ACCU$-preregular in the wider sense of Footnote 5, the rules $R(U)$ are sort-decreasing. Furthermore, the rewrite theory $(\Omega, B, R(U))$ is convergent: it terminates because the rules $R(U)$ are size-decreasing and the axioms $B$ are size-preserving, it is easy to check that it is strictly $B$-coherent, and it is easy to check that it is locally confluent by analysis of critical pairs. This means that we have an isomorphism $T_{\Omega/ACCU} \cong C_{R(U)/B}$, where $C_{R(U)/B}$ denotes the canonical term algebra of $(\Omega, B, R(U))$.

Therefore, deciding whether a sort $s$ is finite in $T_{\Omega/ACCU}$ is equivalent to deciding whether it is finite in $C_{R(U)/B}$. Now consider the following lemma:

**Lemma 3.** *$t \to_{R(U),B} t'$ iff $t \to_{R(U)} t''$ with $t' =_B t''$.*

*Proof.* Since $\to_{R(U)} \subseteq \to_{R(U),B}$, we only need to prove the ($\Rightarrow$) implication. If in a unit rule $f(x, e) \to x$ or $f(e, x) \to x$ the $B$-axioms for $f$ are $C$ or none, the result is trivial. Consider now the case where $f$ is $AC$. Then, if $t \to_{R(U),B} t'$ using one of these two unit rules (in fact, the first one suffices) means that there is a position $p$ in $t$ and a substitution $\sigma = \{x \mapsto v\}$ such that $t|_p =_B f(x, e)\sigma$ and $t' = t[v]_p$. But we can always "flatten" the term $t|_p$ so that $t|_p =_B f(u_1, \ldots, u_n, e)$ and $v =_B f(u_1, \ldots, u_n)$ ($u_1$ if $n = 1$), where the top symbol in each $u_i$ is different from $f$. Furthermore, we can find a position $p.q$ and terms $u_{i_j}$, $1 \leqslant j \leqslant k$, such that $t|_{p.q} = f(w, e)$, or $t|_{p.q} = f(e, w)$, with $w =_B f(u_{i_1}, \ldots, u_{i_k})$, and $t|_p =_B f(u_{l_1}, \ldots, u_{l_m}, f(w, e))$ with $\{1, \ldots, n\} = \{i_1, \ldots, i_k\} \uplus \{l_1, \ldots, l_m\}$, where, by convention, $f(u_{i_1}) = u_{i_1}$ if $k = 1$. But we can perform a rewrite $t \to_{R(U)} t''$ at position $p.q$ with either $f(x, e) \to x$ or $f(e, x) \to x$, and since $t|_p =_B f(u_{l_1}, \ldots, u_{l_m}, f(w, e))$, this means that $t''|_p =_B f(u_{l_1}, \ldots, u_{l_m}, w)$, and therefore that $t' =_B t''$. □

Consider now the convergent rewrite theory $(\Omega, \varnothing, R(U))$, and let $C_{R(U)}$ denote its canonical term algebra. Then, since the above lemma implies that

$t!_{R(U)} =_B t!_{R(U),B}$ for each $t$, we obtain the identity $C_{R(U)/B} = C_{R(U)}/B$. That is, for each sort $s$ we have $C_{R(U)/B,s} = C_{R(U),s}/B$, where $C_{R(U),s}/B$ denotes the set of $B$-equivalence classes of $C_{R(U),s}$. But note that, since the $C$ and $AC$-axioms are symbol- and size-preserving, any $B$-equivalence class $[t]_B$ is a *finite* set. Therefore, $T_{\Omega/ACCU,s}$ is finite iff $C_{R(U)/B,s}$ is finite iff $C_{R(U),s}$ is finite. But the finiteness of $C_{R(U),s}$ is decidable by tree automata techniques, see, e.g., [28], because: (i) $T_{\Omega,s}$ is tree automata definable, since order-sorted signatures are tree automata in disguise, so that $s$ is one of the states of the automaton; (ii) $C_{R(U),s} = T_{\Omega,s} - Red_{R(U)}$, where $Red_{R(U)}$ denotes the set of $R(U)$-reducible terms, and (iii) the rules $R(U)$ are left-linear, making $Red_{R(U)}$ tree automata definable.

Let us now address the question of how, given a finite sort $s$ in $T_{\Omega/ACCU}$, we can effectively compute representatives in the equivalence classes of $T_{\Omega/ACCU,s}$. This is the same as computing representatives in the $B$-equivalence classes of $C_{R(U)/B,s}$. But this is easy if we can effectively generate the set $C_{R(U),s}$, since then we can compare its elements for $B$-equality and choose non-$B$-equal representatives. That we can effectively generate the set $C_{R(U),s}$ follows from the following observations:

1. $C_{R(U),s}$ is tree automata definable, that is, by reasons (i)–(iii) above we can effectively construct a tree automaton $\mathcal{A}$ having a state $q$ such that $C_{R(U),s}$ are the terms accepted by $\mathcal{A}$ with accepting state $q$.
2. The transitions $R$ of the tree automaton $\mathcal{A}$ can be viewed as a ground term rewriting system $(\Sigma(Q), R)$, where $\Sigma(Q)$ is the signature obtained by adding to the unsorted function symbols of $\Sigma$ the states $Q$ of $\mathcal{A}$ as fresh new constants. That is, transitions $f(q_1, \ldots, q_1) \to q$ and epsilon transitions $q \to q'$ are viewed as $\Sigma(Q)$-rewrite rules.
3. Since emptiness of the set of terms recognized by a state $q'$ of $\mathcal{A}$ is decidable, we can, reasoning as in [86], define a smaller tree automaton $\mathcal{A}^\circ$ with states $Q^\circ$ and rules $R^\circ$ where empty states have been removed and $C_{R(U),s}$ is again accepted by state $q$.
4. Consider now the graph containing all terms reachable from $q$ by rewriting with the inverse rules $(R^\circ)^{-1}$. That is, each rule $f(q_1, \ldots, q_1) \to q$, resp. $q \to q'$, is now oriented as $q \to f(q_1, \ldots, q_1)$, resp. $q' \to q$. Note that this graph contains all $t \in C_{R(U),s}$, since $t \in C_{R(U),s}$ iff $s \to!_{(R^\circ)^{-1}} t$. Furthermore, this graph must be *finite*, since otherwise it would contain terms of unbounded size. But since all the states of $Q^\circ$ appearing in such terms recognize nonempty sets of terms, then, replacing each state $q'$ in such a term by a term recognized by $q'$ (which can always be done by further rewriting), we would reach the absurd conclusion that $C_{R(U),s}$ is infinite. Therefore, the terms of $C_{R(U),s}$ can be generated by inspection of this finite graph, as desired.

Note, finally, that a more efficient way of generating the representatives of $T_{\Omega/ACCU,s}$ is to use the $\to_{(R^\circ)^{-1},B}$ rewrite relation and look for terminating nodes in the graph reachable from $q$, since then $B$-equality can be built-in,

so that two graph nodes can be identified iff they are $B$-equal, leading to a potentially much smaller search graph.

## C Proof of Lemma 1

*Proof.* First of all, note that on all such atoms whose left and right sides have both least sorts different from *Elt*, *Set*, *Magma*, or *Pred*, the substitution is the identity function. Since $Set < Elt < Magma$, we need only consider atoms with disequalities between terms of sort *Magma* (or less, but with some side having least sort no lower than *Set*), or between terms of sort *Pred*. To prove the property for all such $t \neq t'$ we reason by strong induction on $n = max(|t|, |t'|)$, where $|t|$ denotes the size of $t$ as a tree.

For the sort *Magma* the base case $n = 1$ must, up to symmetry and disregarding disequalities where both sides have sort $s$ or less, be of one the the following forms: (i) $X \neq \varnothing$, with $X$ either a constant or variable of sort $s$ or less, or a variable of sort *Elt*, *Set* or *Magma*, or (ii) $X \neq Y$, with $X$ either a constant or variable of sort $s$ or less, or a variable of sort *Elt*, *Set* or *Magma*, and $Y$ a variable of sort *Elt*, *Set* or *Magma*, and with $X$ and $Y$ different. In all such cases the substitution yields a normalized and axiom-consistent disequality as claimed. For negative atoms between terms of sort *Pred* the base case $n = 1$ is empty and the property holds trivially.

The proof of the $n + 1$ induction step must consider both the *Magma* and *Pred* cases. Up to $AC \cup B_{(\Omega',\Delta')}$-equality, axiom-consistent and normalized disequalities between terms of sort *Magma* must have the flattened form:

$$\overline{Y}_k, \overline{u}_{k'}, \overline{Y'}_n, \overline{u'}_{n'} \neq \overline{Y}_k, \overline{u}_{k'}, \overline{Y''}_m, \overline{u''}_{m'}$$

where $\overline{Y}_k, \overline{u}_{k'}$ represents the "maximally shared part" between both sides, and where: (i) the $\overline{Y}_k$, $\overline{Y'}_n$ and $\overline{Y''}_m$ are variables of sort *Set*, *Elt* or *Magma*, (ii) the $\overline{u}_{k'}$, $\overline{u'}_{n'}$ and $\overline{u''}_{m'}$ are normalized terms of sort *Set*, or $s$ less, which are not variables of sort *Set*, (iii) all terms in $\overline{Y}_k, \overline{u}_{k'}, \overline{Y'}_n, \overline{u'}_{n'}$ are mutually $AC \cup B_{(\Omega',\Delta')}$-different, and the same holds for all terms in $\overline{Y}_k, \overline{u}_{k'}, \overline{Y''}_m, \overline{u''}_{m'}$, (iv) all terms in $\overline{Y'}_n, \overline{u'}_{n'}\overline{Y''}_m, \overline{u''}_{m'}$ are mutually $AC \cup B_{(\Omega',\Delta')}$-different, and (v) $k + k' \geqslant 0$ and $n + n' + m + m' \geqslant 1$, and if $k + k' = 0$, then $n + n' \geqslant 1$ and $m + m' \geqslant 1$.

Note that if $k + k' > 0$, the induction hypothesis applies to $\overline{Y}_k, \overline{u}_{k''} \neq \overline{Y''}_m, \overline{u''}_{m''}$ and to each of the above-mentioned normalized and axiom-consistent disequalities between individual terms, so the property easily follows. Therefore, we reduce to the case $k + k' = 0$. Then, if $n + n' + m + m' \geqslant 3$, the induction hypothesis applies to each of the above-mentioned normalized and axiom-consistent disequalities between individual terms, so the property easily follows.

This leaves us with the case $n + n' = 1$ and $m + m' = 1$ where, since the base case is already taken care of, we must have $n' + m' \geqslant 1$. Up to symmetry this can be broken into several cases: (i) $\varnothing \neq u$ with $u$ a term of sort $s$ or less, which is left unchanged by the substitution, (ii) $\varnothing \neq \{u\}$, (iii) $\varnothing \neq \{u_1, \ldots, u_n\}$, $n \geqslant 2$,

with $u_i$ $AC \cup B_{(\Omega',\Delta')}$-different from $u_j$ if $i \neq j$, (iv) $X \neq u$, with $X$ a variable of sort *Elt*, *Set* or *Magma* and $u$ of sort $s$ or less, (v) $X \neq \{u\}$, with $X$ a variable of sort *Elt*, *Set* or *Magma*, (vi) $X \neq \{u_1, \ldots, u_n\}$, $n \geqslant 2$, with $X$ a variable of sort *Elt*, *Set* or *Magma* and $u_i$ $AC \cup B_{(\Omega',\Delta')}$-different from $u_j$ if $i \neq j$, or (vi) $\{u_1, \ldots, u_n\} \neq \{v_1, \ldots, v_m\}$ with $n + m \geqslant 2$, $u_i$ $AC \cup B_{(\Omega',\Delta')}$-different from $u_{i'}$ if $i \neq i'$, $v_j$ $AC \cup B_{(\Omega',\Delta')}$-different from $v_{j'}$ if $j \neq j'$, and if $n = m$ some $u_q$ $AC \cup B_{(\Omega',\Delta')}$-different from all the $v_j$. The proof of cases (i) and (iv) is trivial; that of cases (iii) and (vi) follows easily from the induction hypothesis applied to the disequalities between the relevant subterms; that of case (ii) follows from the observation that if $u = \varnothing$ the substitution leaves everything unchanged, and otherwise the induction hypothesis applies to $\varnothing \neq u$, ensuring that $u\{\overline{Y} \mapsto \{\overline{y}\}\}$ is normalized and axiom-consistent. For (iv), a similar case distinction between $X = u$ or $X$ different from $u$ applies to prove the property by ensuring that $u\{\overline{Y} \mapsto \{\overline{y}\}\}$ is normalized and axiom-consistent.

Normalized and axiom-consistent negated atoms of sort *Pred* must be of one of the following forms: (i) $X \subseteq \varnothing \neq tt$, with $X$ a variable of sort *Set*, so that its substitution instance $\{x\} \subseteq \varnothing \neq tt$ is obviously normalized and axiom-consistent; (ii) $\{u\} \subseteq \varnothing \neq tt$, with the $u$ a term of sort *Magma* or less, possibly a variable; then, if $u = \varnothing$ we are done; otherwise the induction hypothesis applies to the disequality $u \neq \varnothing$, so that $u\{\overline{Y} \mapsto \{\overline{y}\}\}$ is normalized and we are again done; (iii) $\{\overline{u}_k\} \subseteq \varnothing \neq tt$, $k \geqslant 2$, with the $u_i$ terms of sort *Magma* or less, including variables of such sorts, where, since all the $u_i$ are normalized, they must be $AC \cup B_{(\Omega',\Delta')}$-different; but then the induction hypothesis applies to the disequalities $u_i \neq u_j$, $i \neq j$, so that all their substitution instances are again normalized and $AC \cup B_{(\Omega',\Delta')}$-different, which makes $\{\overline{u}_k\}\{\overline{Y} \mapsto \{\overline{y}\}\} \subseteq \varnothing$ normalized, as needed for the result to hold; and (iv) (up to $AC \cup B_{(\Omega',\Delta')}$-equality), $\{\overline{u}_k, \overline{v}_n\} \subseteq \{\overline{u}_k, \overline{w}_m\} \neq tt$, with $k \geqslant 0$, $n \geqslant 1$, and if $k = 0$ then $m \geqslant 1$, where all individual terms in $\overline{u}_k, \overline{v}_n, \overline{w}_m$ are mutually $AC \cup B_{(\Omega',\Delta')}$-different. That is, $\overline{u}_k$ is the "maximally shared part" between both sets, which may be empty. Case (iv) breaks into the case where $k, n, m \geqslant 1$, represented by the above "generic" situation, and the two degenerate subcases: (k=0) $\{\overline{v}_n\} \subseteq \{\overline{w}_m\} \neq tt$, $n, m \geqslant 1$, and (m=0) $\{\overline{u}_k, \overline{v}_n\} \subseteq \{\overline{u}_k\} \neq tt$. In all three subcases, the induction hypothesis applied to the mutual disequalities between the individual subterms in (the remaining part of) $\overline{u}_k, \overline{v}_n, \overline{w}_m$ make their substitution instances normalized and mutually $AC \cup B_{(\Omega',\Delta')}$-different. This makes $(\{\overline{u}_k, \overline{v}_n\} \subseteq \{\overline{u}_k, \overline{w}_m\})\{\overline{Y} \mapsto \{\overline{y}\}\}$ normalized and axiom-consistent. □

## D Descending from $\mathcal{Z}_+$ to $\mathcal{N}_+$

Can we drop $\mathcal{Z}_+$'s variant complexity from 12 to 0? We could do so if we show that there is a descent map to $\mathcal{N}_+$. The ideas are well-known (for a sketch see, e.g., [21]). However, the more expressive order-sorted language offers opportunities for making the formula transformations actually simpler, since variables of sort *Nat* or *NzNat* can be left untouched. Given the drastic reduction in variant complexity, the somewhat sketchy presentation in [21], and the fact that I

also include other formula transformations further simplifying the final result, it seems worth giving here a detailed description for the reader's benefit.

To reach $\mathcal{N}_+$ we need to define two descent maps $\mathcal{Z}_+ \xrightarrow{v} \mathcal{Z}_+ \xrightarrow{-} \mathcal{Z}_+$, where $v$ will replace all variables of sorts *Int* or *NzNeg* by corresponding expressions involving variables of sort *NzNat* only; and where '$-$' will replace each term of the form $-(u)$ on one side of an equation or disequation by the term $u$ on the other side. Both descent maps will perform a few additional simplifications to further reduce the size of each resulting conjunction. Since $\varphi^{v\,-}$ will be in the language of $\mathcal{N}_+$, the combined descent map has also the tighter typing $\mathcal{Z}_+ \xrightarrow{v} \mathcal{Z}_+ \xrightarrow{-} \mathcal{N}_+$ and will allows us to reach the desired OS-compact core $\mathcal{N}_+$.

The descent map $\mathcal{Z}_+ \xrightarrow{v} \mathcal{Z}_+$ will first put a QF formula in DNF, normalize it by the rules of $\mathcal{Z}_+$ modulo $ACU$, and remove any $ACU$-inconsistent conjunction. It will also further simplify each equation or disequation in each conjunct by the *cancellation rules* $x + y = x + z \to y = z$, $x + y \neq x + z \to y \neq z$, $x' + y = x' + z \to y = z$, and $x' + y \neq x' + z \to y \neq z$, where $y, z$ have sort *Int*, $x$ sort *NzNat*, and $x'$ sort *NzNeg*. It will then replace *everywhere* within a conjunction each variable $y$ of sort *Int* by the expression $z + -(z')$, with $z, z'$ *fresh* variables of sort *NzNat*. Using a variable $C$ for a conjunction of literals, a variable $D$ for a disjunction of such conjunctions, and assuming that $\vee$ is $ACU$ with identity element $\bot$, this can be achieved using the meta-level rewrite rule, $D \vee C[y] \to D \vee (C\{y \mapsto z + -(z')\})$, where $z, z'$ are *fresh*, the sorts are as stated above, and $C[y]$ abbreviates the occurrence of the variable $y$ somewhere in $C$. Such a rule will be applied repeatedly until no variables of sort *Int* remain in conjunctions. It will likewise replace everywhere within a conjunction each variable $x$ of sort *NzNeg* by the expression $-(x')$, with $x'$ a *fresh* variable of sort *NzNat*. This can be achieved using the meta-level rewrite rule, $D \vee C[x] \to D \vee (C\{x \mapsto -(x')\})$, where $x'$ is *fresh* and the sorts are as stated above. Again, such a rule will be applied repeatedly until no variables of sort *NzNeg* remain in conjunctions. This is indeed a descent map because: (i) any abelian group, and in particular $C_{\mathcal{Z}_+}$, satisfies the equivalence: $x + y = x + z \Leftrightarrow y = z$; (ii) in $C_{\mathcal{Z}_+}$ the conjunctions $C[y]$ and $C\{y \mapsto z + -(z')\}$ (with the assumed sorts for $y, z, z'$ and freshness of $z, z'$) are *equi-satisfiable* (the satisfiability implication ($\Leftarrow$) follows from $C\{y \mapsto z + -(z')\}$ being a substitution instance, and the ($\Rightarrow$) implication from $C_{\mathcal{Z}_+} \models (\forall y)(\exists z, z')\ y = z + -(z')$, with the assumed sorts for $y, z, z'$); and (iii) $C[x]$ and $C\{x \mapsto -(x')\}$ are likewise equisatisfiable (with the assumed sorts for $x, x'$ and freshness of $x'$) by the same reasoning and the fact that $C_{\mathcal{Z}_+} \models (\forall x)(\exists x')\ x = -(x')$, with the assumed sorts for $x, x'$.

The descent map $\mathcal{Z}_+ \xrightarrow{-} \mathcal{Z}_+$ will likewise put a QF formula in DNF, normalize it by the rules of $\mathcal{Z}_+$ modulo $ACU$, remove any $ACU$-inconsistent conjunction, and further simplify each equation or disequation by means of the cancellation rules. It will then eliminate all subterms of the form $-(u)$ in either side of any equation or disequation by repeatedly applying the rewrite rules: $-(x) + y = z \to y = x + z$, $-(x) + y \neq z \to y \neq x + z$, $y = -(x) + z \to x + y = z$, and $y \neq -(x) + z \to x + y \neq z$, where $x$ has sort *NzNat* and $y, z$ sort *Int*. It will finally further simplify each conjunction by applying the cancellation rules.

This is also a descent map, because $C_{\mathcal{Z}_+} \models -(x) + y = z \Leftrightarrow y = x + z$, and $C_{\mathcal{Z}_+} \models y = -(x) + z \Leftrightarrow x + y = z$, again, with the assumed sorts for $x, y, z$.

It is easy to show that $\varphi^{v\,-}$ is in the language of $\mathcal{N}_+$, so that, since $\mathcal{Z}_+$ protects $\mathcal{N}_+$, we have $C_{\mathcal{Z}_+} \models \varphi^{v\,-}$ iff $C_{\mathcal{N}_+} \models \varphi^{v\,-}$. Therefore, we have a descent map $\mathcal{Z}_+ \xrightarrow{v} \mathcal{Z}_+ \xrightarrow{-} \mathcal{N}_+$, making $\mathcal{N}_+$ an OS-compact core for $\mathcal{Z}_+$.