Variant-Based Decidable Satisfiability in Initial Algebras with Predicates*

Raúl Gutiérrez[1] and José Meseguer[2]

[1] Universitat Politècnica de València
[2] University of Illinois at Urbana-Champaign

Abstract. Decision procedures can be either *theory-specific*, e.g., Presburger arithmetic, or *theory-generic*, applying to an infinite number of user-definable theories. Variant satisfiability is a theory-generic procedure for quantifier-free satisfiability in the initial algebra of an order-sorted equational theory $(\Sigma, E \cup B)$ under two conditions: (i) $E \cup B$ has the *finite variant property* and B has a finitary unification algorithm; and (ii) $(\Sigma, E \cup B)$ protects a constructor subtheory $(\Omega, E_\Omega \cup B_\Omega)$ that is OS-*compact*. These conditions apply to many user-definable theories, but have a main limitation: they apply well to *data structures*, but often do *not* hold for user-definable *predicates* on such data structures. We present a theory-generic satisfiability decision procedure, and a prototype implementation, extending variant-based satisfiability to initial algebras with user-definable predicates under fairly general conditions.
Keywords: finite variant property (FVP), OS-compactness, user-definable predicates, decidable validity and satisfiability in initial algebras.

1 Introduction

Some of the most important recent advances in software verification are due to the systematic use of decision procedures in both model checkers and theorem provers. However, a key limitation in exploiting the power of such decision procedures is their current *lack of extensibility*. The present situation is as follows. Suppose a system has been formally specified as a theory T about which we want to verify some properties, say $\varphi_1, \ldots, \varphi_n$, using some model checker or theorem prover that relies on an SMT solver for its decision procedures. This limits *a priori* the decidable subtheory $T_0 \subseteq T$ that can be handled by the SMT solver. Specifically, the SMT solver will typically support a fixed set $Q_1, \ldots, Q_k$ of decidable theories, so that, using a theory combination method such as Nelson and Oppen [26], or Shostak [27], T_0 must be a finite combination of the decidable theories $Q_1, \ldots, Q_k$ supported by the SMT solver.

* Partially supported by NSF Grant CNS 14-09416, NRL under contract number N00173-17-1-G002, the EU (FEDER), Spanish MINECO project TIN2015-69175-C4-1-R and GV project PROMETEOII/2015/013. Raúl Gutiérrez was also supported by INCIBE program "Ayudas para la excelencia de los equipos de investigación avanzada en ciberseguridad".

In non-toy applications it is unrealistic to expect that the entire specification T of a software system will be decidable. Obviously, the bigger the decidable subtheory $T_0 \subseteq T$, the higher the levels of automation and the greater the chances of scaling up the verification effort. With *theory-specific* procedures for, say, $Q_1, \ldots, Q_k$, the decidable fragment T_0 of T is a priori bounded. One promising way to extend the decidable fragment T_0 is to develop *theory-generic* satisfiability procedures. These are procedures that make decidable not a single theory Q, but an *infinite class* of *user-specifiable* theories. Therefore, an SMT solver supporting both theory-specific and theory-generic decision procedures becomes *user-extensible* and can carve out a potentially much bigger Decidable Fragment T_0 of the given system specification T.

Variant-based satisfiability [22,21] is a recent theory-generic decision procedure applying to the following, easily user-specifiable infinite class of equational theories $(\Sigma, E \cup B)$: (i) Σ is an order-sorted [15] signature of function symbols, supporting types, subtypes, and subtype polymorphisms; (ii) $E \cup B$ has the *finite variant property* [9] and B has a finitary unification algorithm; and (iii) $(\Sigma, E \cup B)$ protects a constructor subtheory $(\Omega, E_\Omega \cup B_\Omega)$ that is OS-*compact* [22,21]. The procedure can then decide satisfiability in the initial algebra $T_{\Sigma/E \cup B}$, that is, in the *algebraic data type* specified by $(\Sigma, E \cup B)$. These conditions apply to many user-definable theories, but have a main limitation: they apply well to *data structures*, but often do *not* hold for user-definable *predicates*.

The notions of variant and of OS-compactness mentioned above are defined in detail in Section 2. Here we give some key intuitions. Given Σ-equations $E \cup B$ such that the equations E oriented as left-to-right rewrite rules are confluent and terminating modulo the equational axioms B, a *variant* of a Σ-term t is a pair (u, θ) where θ is a substitution, and u is the canonical form of the term instance $t\theta$ by the rewrite rules E modulo B. Intuitively, the variants of t are the fully simplified *patterns* to which the instances of t can reduce. Some simplified instances are of course more general (as patterns) than others. $E \cup B$ has the *finite variant property* (FVP) if any Σ-term t has a *finite* set of most general variants. For example, the addition equations $E = \{x + 0 = x, x + s(y) = s(x + y)\}$ are *not* FVP, since $(x + y, id)$, $(s(x + y_1), \{y \mapsto s(y_1)\})$, $(s(s(x + y_2)), \{y \mapsto s(s(y_2))\})$, …, $(s^n(x + y_n), \{y \mapsto s^n(y_n)\})$, …, are all *incomparable* variants of $x + y$. Instead, the Boolean equations $G = \{x \vee \top = \top, x \vee \bot = x, x \wedge \top = x, x \wedge \bot = \bot\}$ *are* FVP. For example, the most general variants of $x \vee y$ are: $(x \vee y, id)$, $(x, \{y \mapsto \bot\})$, and $(\top, \{y \mapsto \top\})$. Assuming for simplicity that all sorts in a theory $(\Omega, E_\Omega \cup B_\Omega)$ have an infinite number of ground terms of that sort which are all different modulo the equations $E_\Omega \cup B_\Omega$, then OS-compactness of $(\Omega, E_\Omega \cup B_\Omega)$ means that any conjunction of disequalities $\bigwedge_{1 \leqslant i \leqslant n} u_i \neq v_i$ such that $E_\Omega \cup B_\Omega \not\vdash u_i = v_i$, $1 \leqslant i \leqslant n$, is *satisfiable* in the initial algebra $T_{\Omega/E_\Omega \cup B_\Omega}$, obtaining a decision procedure. For example, $(\{0, s\}, \varnothing)$ is OS-compact, where $\{0, s\}$ are the usual natural number constructors. Thus, $s(x) \neq s(y) \wedge 0 \neq y$ *is* satisfiable in $T_{\{0,s\}}$.

The key reason why user-definable predicates present a serious obstacle is the following. Variant satisfiability works by *reducing* satisfiability in the initial algebra $T_{\Sigma/E \cup B}$ to satisfiability in the much simpler algebra of construc-

tors $T_{\Omega/E_\Omega \cup B_\Omega}$. In many applications $E_\Omega = \varnothing$, and if the axioms B_Ω are any combination of associativity, commutativity and identity axioms, except associativity without commutativity, then (Ω, B_Ω) is an OS-compact theory [22,21], making satisfiability in T_{Ω/B_Ω} and therefore in $T_{\Sigma/E \cup B}$ decidable. We can equationally specify a predicate p with sorts $A_1, \ldots, A_n$ in a *positive* way as a function $p : A_1, \ldots, A_n \to \mathit{Pred}$, where the sort *Pred* of predicates contains a "true" constant tt, so that $p(u_1, \ldots, u_n)$ not holding for concrete ground arguments $u_1, \ldots, u_n$ is expressed as the *disequality* $p(u_1, \ldots, u_n) \neq tt$. But $p(u_1, \ldots, u_n) \neq tt$ means that p must be a *constructor* of sort *Pred* in Ω, and that the equations defining p must belong to E_Ω, making $E_\Omega \neq \varnothing$ and ruling out the case when $T_{\Omega/E_\Omega \cup B_\Omega} = T_{\Omega/B_\Omega}$ is decidable by OS-compactness.

This work extends variant-based satisfiability to initial algebras with user-definable predicates under fairly general conditions using two key ideas: (i) characterizing the cases when $p(u_1, \ldots, u_n) \neq tt$ by means of *constrained patterns*; and (ii) eliminating all occurrences of disequalities of the form $p(v_1, \ldots, v_n) \neq tt$ in a quantifier-free (QF) formula by means of such patterns. In this way, the QF satisfiability problem can be reduced to formulas involving only *non-predicate* constructors, for which OS-compactness holds in many applications. More generally, if some predicates fall within the OS-compact fragment, they can be kept.

Preliminaries are in Section 2. Constructor variants and OS-compactness in Section 3. The satisfiability decision procedure is defined and proved correct in Section 4, and its prototype implementation is described in Section 5. Related work and conclusions are discussed in Section 6. All proofs can be found in [16].

2 Many-Sorted Logic, Rewriting, and Variants

We present some preliminaries on many-sorted (MS) logic, rewriting and finite variant and variant unification notions needed in the paper. For a more general treatment using order-sorted (OS) logic see [16].

We assume familiarity with the following basic concepts and notation that are explained in full detail in, e.g., [24]: (i) *many-sorted (MS) signature* as a pair $\Sigma = (S, \Sigma)$ with S a set of *sorts* and Σ an $S^* \times S$-indexed family $\Sigma = \{\Sigma_{w,s}\}_{(w,s) \in S^* \times S}$ of function symbols, where $f \in \Sigma_{s_1 \ldots s_n, s}$ is displayed as $f : s_1 \ldots s_n \to s$; (ii) *Σ-algebra* A as a pair $A = (A, {}_{-A})$ with $A = \{A_s\}_{s \in S}$ an S-indexed family of sets, and ${}_{-A}$ a mapping interpreting each $f : s_1 \ldots s_n \to s$ as a function in the set $[A_{s_1} \times \ldots \times A_{s_n} \to A_s]$. (iii) *$\Sigma$-homomorphism* $h : A \to B$ as an S-indexed family of functions $h = \{h_s : A_s \to B_s\}_{s \in S}$ preserving the operations in Σ; (iv) the Σ-algebra T_Σ and its initiality in the category $\mathbf{MSAlg}_\Sigma$ of Σ-algebras when Σ is unambiguous.

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

of $T_\Sigma(X)$ means that for each $A \in \mathbf{MSAlg}_\Sigma$ and $\alpha \in [X \to A]$, there exists a unique Σ-homomorphism, $\_\alpha : T_\Sigma(X) \to A$ extending α, 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 $A = T_\Sigma(Y)$, an interpretation of the constants in X, i.e., an S-sorted function $\sigma \in [X \to T_\Sigma(Y)]$ is called a *substitution*, and its unique homomorphic extension $\_\sigma : T_\Sigma(X) \to T_\Sigma(Y)$ is also called a substitution. Define $dom(\sigma) = \{x \in X \mid x \neq x\sigma\}$, and $ran(\sigma) = \bigcup_{x \in dom(\sigma)} vars(x\sigma)$. Given variables Z, the substitution $\sigma|_Z$ agrees with σ on Z and is the identity elsewhere.

We also assume familiarity with many-sorted first-order logic including: (i) the first-order language of Σ-*formulas* for Σ a signature (in our case Σ has only function symbols and the $=$ predicate); (ii) given a Σ-algebra A, a formula $\varphi \in Form(\Sigma)$, and an assignment $\alpha \in [Y \to A]$, with $Y = fvars(\varphi)$ the free variables of φ, the *satisfaction relation* $A, \alpha \models \varphi$; (iii) the notions of a formula $\varphi \in Form(\Sigma)$ being *valid*, denoted $A \models \varphi$, resp. *satisfiable*, in a Σ-algebra A. For a subsignature $\Omega \subseteq \Sigma$ and $A \in \mathbf{MSAlg}_\Sigma$, the *reduct* $A|_\Omega \in \mathbf{MSAlg}_\Omega$ agrees with A in the interpretation of all sorts and operations in Ω and discards everything in $\Sigma \backslash \Omega$. If $\varphi \in Form(\Omega)$ we have the equivalence $A \models \varphi \;\Leftrightarrow\; A|_\Omega \models \varphi$.

An MS *equational theory* is a pair $T = (\Sigma, E)$, with E a set of Σ-equations. $\mathbf{MSAlg}_{(\Sigma,E)}$ denotes the full subcategory of $\mathbf{MSAlg}_\Sigma$ with objects those $A \in \mathbf{MSAlg}_\Sigma$ such that $A \models E$, called the (Σ, E)-*algebras*. $\mathbf{MSAlg}_{(\Sigma,E)}$ has an *initial algebra* $T_{\Sigma/E}$ [24]. The inference system in [24] is *sound and complete* for MS equational deduction, i.e., for any MS equational theory (Σ, E), and Σ-equation $u = v$ we have an equivalence $E \vdash u = v \;\Leftrightarrow\; E \models u = v$. For the sake of simpler inference we assume *non-empty sorts*, i.e., $\forall s \in S,\ T_\Sigma, s \neq \varnothing$. Deducibility $E \vdash u = v$ is abbreviated as $u =_E v$.

In the above notions there is only an *apparent* lack of predicate symbols: full many-sorted first-order logic can be *reduced* to many-sorted algebra and the above language of equational formulas. The reduction is achieved as follows. A many-sorted first-order (MS-FO) signature, is a pair (Σ, Π) with Σ a MS signature with set of sorts S, and Π an S^*-indexed set $\Pi = \{\Pi_w\}_{w \in S^*}$ of *predicate symbols*. We associate to a MS-FO signature (Σ, Π) a MS signature $(\Sigma \cup \Pi)$ by adding to Σ a new sort *Pred* with a constant tt and viewing each $p \in \Pi_w$ as a function symbol $p : s_1 \dots s_n \to Pred$. The reduction at the model level is now very simple: each $(\Sigma \cup \Pi)$-algebra A defines a (Σ, Π)-model A° with Σ-algebra structure $A|_\Sigma$ and having for each $p \in \Pi_w$ the predicate interpretation $A^\circ_p = A^{-1}_{p:w \to Pred}(tt)$. The reduction at the formula level is also quite simple: we map a (Σ, Π)-formula φ to an equational formula $\widetilde{\varphi}$, called its *equational version*, by just replacing each atom $p(t_1, \dots, t_n)$ by the equational atom $p(t_1, \dots, t_n) = tt$. The *correctness* of this reduction is just the easy to check equivalence:

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

A MS-FO *theory* is just a pair $((\Sigma, \Pi), \Gamma)$, with (Σ, Π) a MS-FO signature and Γ a set of (Σ, Π)-formulas. Call $((\Sigma, \Pi), \Gamma)$ *equational* iff $(\Sigma \cup \Pi, \widetilde{\Gamma})$ is a many-sorted equational theory. By the above equivalence and the completeness of many-sorted 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 many-sorted Horn Logic with Equality and therefore has an initial model $T_{(\Sigma,\Pi),\Gamma}$ [14]. A useful, easy to check fact is that we have an identity: $T^{\circ}_{\Sigma \cup \Pi/\tilde{\Gamma}} = T_{(\Sigma,\Pi),\Gamma}$.

Recall the notation for term positions, subterms, and term replacement from [10]: (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 1. *A* rewrite theory *is a triple* $\mathcal{R} = (\Sigma, B, R)$ *with* (Σ, B) *a MS equational theory and* R *a set of* Σ-rewrite rules*, i.e., sequents* $l \to r$*, with* $l, r \in T_\Sigma(X)_s$ *for some* $s \in S$*. 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 equation* $u = v \in B$ *is* regular*, i.e.,* $vars(u) = vars(v)$*, and* linear*, i.e., there are no repeated variables in either* u *or* v*. The* one-step R, B*-rewrite relation* $t \to_{R,B} t'$*, holds between* $t, t' \in T_\Sigma(X)_s$*,* $s \in S$*, 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* $t' = t[r\sigma]_p$*.*

$\mathcal{R}$ *is called: (i)* terminating *iff the relation* $\to_{R,B}$ *is well-founded; (ii)* strictly B-coherent *[23] 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* Σ*-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* $t \to^!_{R,B} t!_{R,B}$*, 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 Σ-equations, let $R(E) = \{u \to v \mid u = v \in E\}$. A *decomposition* of a MS equational theory (Σ, 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 1. *(Church-Rosser Theorem) [17,23] Let* $\mathcal{R} = (\Sigma, B, R)$ *be a decomposition of* (Σ, E)*. Then we have an equivalence:*

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

If $\mathcal{R} = (\Sigma, B, R)$ is a decomposition of (Σ, 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 $f : s_1 \dots s_n \to s$ 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 Σ 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 Ω-terms, whereas the function symbols $f \in \Delta$ are viewed as *defined functions* which are *evaluated away* by R, B-simplification. Ω (with same poset of sorts as Σ) is then called a *constructor subsignature* of Σ. Call a decomposition $\mathcal{R} = (\Sigma, B, R)$ of (Σ, E) *sufficiently complete* with respect to the *constructor subsignature* Ω 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 $[u]_B \subseteq T_\Omega$. We will give several examples of decompositions $\Sigma = \Omega \uplus \Delta$ into constructors and defined functions.

As we can see in the following definition, sufficient completeness is closely related to the notion of a *protecting* theory inclusion.

Definition 2. *An equational theory* (Σ, E) protects *another theory* (Ω, E_Ω) *iff* $(\Omega, E_\Omega) \subseteq (\Sigma, E)$ *and the unique* Ω*-homomorphism* $h : T_{\Omega/E_\Omega} \to T_{\Sigma/E}|_\Omega$ *is an isomorphism* $h : T_{\Omega/E_\Omega} \cong T_{\Sigma/E}|_\Omega$. *A decomposition* $\mathcal{R} = (\Sigma, B, R)$ protects *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)* $C_{\mathcal{R}_0} = C_\mathcal{R}|_{\Sigma_0}$.

$\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ *is a* constructor decomposition *of* $\mathcal{R} = (\Sigma, B, R)$ *iff* $\mathcal{R}$ *protects* $\mathcal{R}_\Omega$ *and,* Σ *and* Ω *have the same poset of sorts, so that by (iii) above* $\mathcal{R}$ *is sufficiently complete with respect to* Ω. *Furthermore,* Ω *is called a subsignature of* free constructors modulo B_Ω *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 Ω being a subsignature of free constructors modulo B_Ω. 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 $C_{\mathcal{R}_0}$.

Given a MS equational theory (Σ, E) and a conjunction of Σ-equations $\phi = u_1 = v_1 \wedge \ldots \wedge u_n = v_n$, an *E-unifier* of ϕ is a substitution σ such that $u_i\sigma =_E v_i\sigma$, $1 \leqslant i \leqslant n$. An *E-unification algorithm* for (Σ, E) is an algorithm generating for each system of Σ-equations ϕ and finite set of variables $W \supseteq vars(\phi)$ a *complete set* of E-unifiers $Unif_E^W(\phi)$ where each $\tau \in Unif_E^W(\phi)$ is assumed idempotent and with $dom(\tau) = vars(\phi)$, and is "away from W" in the sense that $ran(\tau) \cap W = \emptyset$. The set $Unif_E^W(\phi)$ is called "complete" in the precise sense that for any E-unifier σ of ϕ there is a $\tau \in Unif_E(\phi)$ and a substitution ρ such that $\sigma|_W =_E (\tau\rho)|_W$, where, by definition, $\alpha =_E \beta$ means $(\forall x \in X)\ \alpha(x) =_E \beta(x)$ for substitutions α, β. Such an algorithm is called *finitary* if it always terminates with a *finite set* $Unif_E^W(\phi)$ for any ϕ.

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 3. *Given a decomposition* $\mathcal{R} = (\Sigma, B, R)$ *of a MS equational theory* (Σ, E) *and a* Σ*-term* t, *a* variant[3] *[9,13] of* t *is a pair* (u, θ) *such that: (i)* $u =_B$

[3] For a discussion of similar but not exactly equivalent versions of the variant notion see [6]. Here we follow the shaper formulation in [13], rather than the one in [9], because it is technically essential for some results to hold [6].

$(t\theta)!_{R,B}$, *(ii)* $dom(\theta) \subseteq vars(t)$, *and (iii)* $\theta = \theta!_{R,B}$, *that is,* $\theta(x) = \theta(x)!_{R,B}$ *for all variables* x. (u, θ) *is called a* ground variant *iff, furthermore,* $u \in T_\Sigma$. *Given variants* (u, θ) *and* (v, γ) *of* t, (u, θ) *is called* more general *than* (v, γ), *denoted* $(u, \theta) \sqsupseteq_B (v, \gamma)$, *iff there is a substitution* ρ *such that: (i)* $(\theta\rho)|_{vars(t)} =_B \gamma$, *and (ii)* $u\rho =_B v$. *Let* $[\![t]\!]_{R,B} = \{(u_i, \theta_i) \mid i \in I\}$ *denote a* complete set of variants *of* t, *that is, a set of variants such that for any variant* (v, γ) *of* t *there is an* $i \in I$, *such that* $(u_i, \theta_i) \sqsupseteq_B (v, \gamma)$.

A decomposition $\mathcal{R} = (\Sigma, B, R)$ *of* (Σ, E) *has the* finite variant property *[9] (FVP) iff for each* Σ*-term* t *there is a* finite *complete set of variants* $[\![t]\!]_{R,B} = \{(u_1, \theta_1), \ldots, (u_n, \theta_n)\}$. *If* B *has a finitary* B*-unification algorithm the relation* $(u, \alpha) \sqsupseteq_B (v, \beta)$ *is decidable by* B*-matching. Under this assumption on* B, *if* $\mathcal{R} = (\Sigma, B, R)$ *is FVP,* $[\![t]\!]_{R,B}$ *can be chosen to be not only complete, but also a set of* most general *variants, in the sense that for* $1 \leqslant i < j \leqslant n$, $(u_i, \theta_i) \not\sqsupseteq_B (u_j, \theta_j) \wedge (u_j, \theta_j) \not\sqsupseteq_B (u_i, \theta_i)$. *Also, given any finite set of variables* $W \supseteq vars(t)$ *we can always choose* $[\![t]\!]_{R,B}$ *to be of the form* $[\![t]\!]^W_{R,B}$, *where each* $(u_i, \theta_i) \in [\![t]\!]^W_{R,B}$ *has* θ_i *idempotent with* $dom(\theta_i) = vars(t)$, *and "away from* W*," in the sense that* $ran(\theta_i) \cap W = \varnothing$.

If B has a finitary unification algorithm, the *folding variant narrowing* strategy described in [13] provides an effective method to generate $[\![t]\!]_{R,B}$. Furthermore, folding variant narrowing *terminates* for each input $t \in T_\Sigma(X)$ with a finite set $[\![t]\!]_{R,B}$ iff $\mathcal{R}$ has FVP [13].

Two example theories, one FVP and another not FVP, were given in the Introduction. Many other examples are given in [22]. The following will be used as a running example of an FVP theory:

Example 1. (Sets of Natural Numbers). Let $NatSet = (\Sigma, B, R)$ be the following equational theory. Σ has sorts Nat and $NatSet$, subsort inclusion[4] $Nat < NatSet$, and decomposes as $\Sigma = \Omega_c \uplus \Delta$, where the constructors Ω_c include the following operators: 0 and 1 of sort Nat, $\_ + \_ : Nat\ Nat \to Nat$ (addition), $\varnothing$ of sort $NatSet$ and $\_,\_ : NatSet\ NatSet \to NatSet$ (set union). B decomposes as $B = B_{\Omega_c} \uplus B_\Delta$. The axioms B_{Ω_c} include: (i) the associativity and commutativity of $\_ + \_$ with identity 0, the associativity and commutativity of $\_,\_$. R decomposes as $R = R_{\Omega_c} \uplus R_\Delta$. The rules R_{Ω_c} include: (i) an identity rule for union $NS, \varnothing \to NS$; and (ii) idempotency rules for union $NS, NS \to NS$, and $NS, NS, NS' \to NS, NS'$. The signature Δ of defined functions has operators $max : Nat\ Nat \to Nat$, $min : Nat\ Nat \to Nat$, and $\_ \dot{-} \_ : Nat\ Nat \to Nat$, for the maximum, minimum and "monus" (subtraction) functions. The axioms B_Δ are the commutativity of the max and min functions. The rules R_Δ for the defined functions are: $max(N, N + M) \to N + M$, $min(N, N + M) \to N$, $N \dot{-} (N + M) \to 0$, and $(N + M) \dot{-} N \to M$, where N and M have sort Nat.

FVP is a *semi-decidable* property [6], which can be easily verified (when it holds) by checking (using folding variant narrowing supported by Maude 2.7)

[4] As pointed out at the beginning of Section 2, [16] treats the more general *order-sorted* case, where sorts form a poset $(S, \leqslant)$ with $s \leqslant s'$ interpreted as set containment $A_s \subseteq A_{s'}$ in a Σ-algebra A. All results in this paper hold in the order-sorted case.

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. For example, the variant complexity of *NatSet* above is 16.

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 an MS signature Σ with sorts S to an OS signature $\Sigma^\wedge$ by: (1) adding to S fresh new sorts *Lit* and *Conj* with a subsort inclusion $Lit < Conj$; (2) adding a binary conjunction operator $\_\wedge\_ : Lit\ Conj \to Conj$; and (3) adding for each $s \in S$ binary operators $\_ = \_ : s\ s \to Lit$ and $\_ \neq \_ : s\ s \to Lit$.

Variant-based unification goes back to [13]. The paper [22] gives a more precise characterization using $\Sigma^\wedge$-terms as follows. If $\mathcal{R} = (\Sigma, B, R)$ is an FVP decomposition of (Σ, E) and B has a finitary B-unification algorithm, given a system of Σ-equations ϕ with variables W, folding variant narrowing computes a *finite* set $VarUnif_E^W(\phi)$ of E-unifiers away from W that is *complete* in the strong sense that if α is an R, B-normalized E-unifier of ϕ there exists $\theta \in VarUnif_E^W(\phi)$ and an R, B-*normalized* ρ such that $\alpha|_W =_B (\theta\rho)|_W$.

3 Constructor Variants and OS-Compactness

We gather some technical notions and results needed for the inductive satisfiability procedure given in Section 4.

The notion of *constructor variant* is used to answer the question: what variants of t cover as instances modulo B_Ω all canonical forms of all ground instances of t? The following lemma (stated and proved at the more general order-sorted level in [16], but stated here for the MS case for simplicity) gives a precise answer under reasonable assumptions. For more on constructor variants see [22,28,16].

Lemma 1. *Let $\mathcal{R} = (\Sigma, B, R)$ be an FVP decomposition of (Σ, E) protecting a constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$. Assume that: (i) $\Sigma = \Omega \cup \Delta$ with $\Omega \cap \Delta = \varnothing$; (ii) B has a finitary B-unification algorithm and $B = B_\Omega \uplus B_\Delta$, with B_Ω Ω-equations and if $u = v \in B_\Delta$, u, v are non-variable Δ-terms. Call $[\![t]\!]^\Omega_{R,B} = \{(v, \theta) \in [\![t]\!]_{R,B} \mid v \in T_\Omega(X)\}$ the set of* constructor variants *of t. If $[u] \in \mathcal{C}_{\mathcal{R}_\Omega}$ is of the form $u =_B (t\gamma)!_{R,B}$, then there is $(v, \theta) \in [\![t]\!]^\Omega_{R,B}$ and a normalized ground substitution τ such that $u =_B v\tau$.*

We finally need the notion of an order-sorted OS-*compact* equational OS-FO theory $((\Sigma, \Pi), \Gamma)$, generalizing the compactness notion in [8]. The notion is the *same* (but called MS-compactness) for the special case of MS theories treated in the preliminaries to simplify the exposition. It is stated here in the more general OS case because the satisfiability algorithm in Section 4 works for the more general OS case, and the paper's examples are in fact OS theories.

Given a OS equational theory (Σ, E), call a Σ-equality $u = v$ *E-trivial* iff $u =_E v$, and call a Σ-disequality $u \neq v$, denoting the negated atom $\neg(u = v)$,

E-consistent iff $u \neq_E v$. Likewise, call a conjunction $\bigwedge D$ of Σ-disequalities *E-consistent* iff each $u \neq v$ in D is so. Call a sort $s \in S$ *finite* in both (Σ, E) and $T_{\Sigma/E}$ iff $T_{\Sigma/E,s}$ is a finite set, and *infinite* otherwise.

Definition 4. *An equational OS-FO theory* $((\Sigma, \Pi), \Gamma)$ *is called* OS-compact *iff: (i) for each sort* s *in* Σ *we can effectively determine whether* s *is finite or infinite in* $T_{\Sigma \cup \Pi/\tilde{\Gamma},s}$*, and, if finite, can effectively compute a representative ground term* $rep([u]) \in [u]$ *for each* $[u] \in T_{\Sigma \cup \Pi/\tilde{\Gamma},s}$*; (ii)* $=_{\tilde{\Gamma}}$ *is decidable and* $\widetilde{\Gamma}$ *has a finitary unification algorithm; and (iii) any finite conjunction* $\bigwedge D$ *of negated* (Σ, Π)*-atoms whose variables all have infinite sorts and such that* $\bigwedge \widetilde{D}$ *is* $\widetilde{\Gamma}$*-consistent is satisfiable in* $T_{\Sigma,\Pi,\Gamma}$*.*

Call an OS theory (Σ, E) *OS-*compact *iff OS-FO theory* $((\Sigma, \varnothing), E)$ *is OS-*compact*.*

The key theorem, generalizing a similar one in [8] is the following:

Theorem 2. *[22,21] If* $((\Sigma, \Pi), \Gamma)$ *is an* OS-compact *theory, then satisfiability of QF* (Σ, Π)*-formulas in* $T_{\Sigma,\Pi,\Gamma}$ *is decidable.*

The following OS-compactness results are proved in detail in [22]: (i) a free constructor decomposition modulo axioms $\mathcal{R}_\Omega = (\Omega, B_\Omega, \varnothing)$ for B_Ω any combination of associativity, commutativity and identity axioms, except associativity without commutativity, is OS-compact; and (ii) the constructor decompositions for parameterized modules for lists, compact lists, multisets, sets, and hereditarily finite (HF) sets are all *OS-compact-preserving*, in the sense that if the actual parameter has an OS-compact constructor decomposition, then the corresponding instantiation of the parameterized constructor decomposition is OS-compact.

Example 2. The constructor decomposition $\mathcal{R}_{\Omega_c} = (\Omega, B_{\Omega_c}, R_{\Omega_c})$ for the *NatSet* theory in Example 1 is OS-compact. This follows from the fact that *NatSet* is just the instantiation of the constructor decomposition for the parameterized module of (finite) sets in [22] to the natural numbers with 0, 1, and $\_ + \_$, which is itself a theory of free constructors modulo associativity, commutativity and identity 0 for $\_ + \_$ and therefore OS-compact by (i), so that, by (ii), $\mathcal{R}_{\Omega_c} = (\Omega, B_{\Omega_c}, R_{\Omega_c})$ is also OS-compact.

4 QF Satisfiability in Initial Algebras with Predicates

The known variant-based quantifier-free (QF) satisfiability and validity results [22,21] apply to the initial algebra $T_{\Sigma/E}$ of an equational theory (Σ, E) such that (1) $\mathcal{R} = (\Sigma, B, R)$ is a FVP variant-decomposition, (2) $\mathcal{R}$ protects a constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ and (3): (i) B has a finitary unification algorithm; and (ii) the equational theory of $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ is OS-compact.

Example 3. QF validity and satisfiability in the initial algebra $T_{\Sigma/E}$ for (Σ, E) the theory with the *NatSet* FVP variant-decomposition $\mathcal{R} = (\Sigma, B, R)$ in Example 1 are decidable because its axioms B have a finitary unification algorithm and, as explained in Example 2, its constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ is OS-compact.

The decidable inductive validity and satisfiability results in [22,21] apply indeed to many *data structures* of interest, which may obey structural axioms B such as commutativity, associativity-commutativity, or identity. Many useful examples are given in [22], and a prototype Maude implementation is presented in [28]. There is, however, a main limitation about the range of examples to which these results apply, which this work directly addresses. The limitation comes from the introduction of *user-definable predicates*. Recall that we represent a predicate p with sorts $s_1, \ldots, s_n$ as a function $p : s_1, \ldots, s_n \rightarrow \mathit{Pred}$ defined in the *positive* case by confluent and terminating equations $p(u_1^i, \ldots, u_n^i) = tt$, $1 \leqslant i \leqslant k$. The key problem with such predicates p is that, except in trivial cases, there are typically ground terms $p(v_1, \ldots, v_n)$ for which the predicate does *not* hold. This means that p must be a *constructor* operator of sort *Pred* which is *not* a free constructor modulo the axioms B_Ω. This makes proving OS-compactness for a constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ including user-definable predicates a non-trivial case-by-case task. For example, the proofs of OS-compactness for the set containment predicate $\_ \subseteq \_$ in the parameterized module of finite sets and for other such predicates in other FVP parameterized modules in [22] all required non-trivial analyses. Furthermore, OS-compactness may fail for some $\mathcal{R}_\Omega$ precisely because of predicates (see Example 4 below).

Example 4. Consider the following extension by predicates *NatSetPreds* of the *NatSet* theory in Example 1. Its constructor signature $\Omega = \Omega_c \uplus \Omega_\Pi$ adds the subsignature Ω_Π containing the sort *Pred*, a constant tt of sort *Pred*, the subset containment predicate $\_ \subseteq \_ : \mathit{NatSet}\ \mathit{NatSet} \rightarrow \mathit{Pred}$, the strict order predicate $\_ > \_ : \mathit{Nat}\ \mathit{Nat} \rightarrow \mathit{Pred}$, the "sort predicate" $\_ : \mathit{Nat} : \mathit{NatSet} \rightarrow \mathit{Pred}$, characterizing when a set of natural numbers is a natural, and the even and odd predicates $\mathit{even}, \mathit{odd} : \mathit{NatSet} \rightarrow \mathit{Pred}$, defined by the rules R_Π: $\varnothing \subseteq NS \rightarrow tt$, $NS \subseteq NS \rightarrow tt$, $NS \subseteq NS, NS' \rightarrow tt$, $N + M + 1 > N \rightarrow tt$, $N{:}\mathit{Nat} \rightarrow tt$, $\mathit{even}(N + N) \rightarrow tt$ and $\mathit{odd}(N + N + 1) \rightarrow tt$, where NS and NS' have sort *NatSet*, and N and M have sort *Nat*. *NatSetPreds* is FVP, but its constructor decomposition $\mathcal{R}_\Omega = (\Omega_c \uplus \Omega_\Pi, B_{\Omega_c}, R_{\Omega_c} \uplus R_\Pi)$ is *not* OS-compact, since the negation of the trichotomy law $N > M \vee M > N \vee N = M$ is the B_{Ω_c}-consistent but *unsatisfiable* conjunction $N > M \neq tt \wedge M > N \neq tt \wedge N \neq M$.

The goal of this work is to provide a decision procedure for validity and satisfiability of QF formulas in the initial algebra of an FVP theory $\mathcal{R}$ that may contain user-definable predicates and protects a constructor decomposition $\mathcal{R}_\Omega$ that need not be OS-compact, under the following reasonable assumptions: (1) $\mathcal{R} = (\Delta \uplus \Omega_c \uplus \Omega_\Pi, B_\Delta \uplus B_{\Omega_c}, R_\Delta \uplus R_{\Omega_c} \uplus R_\Pi)$ protects $\mathcal{R}_\Omega = (\Omega_c \uplus \Omega_\Pi, B_{\Omega_c}, R_{\Omega_c} \uplus R_\Pi)$, where Ω_Π consists only of predicates, and R_Π consists of rules of the form $p(u_1^i, \ldots, u_n^i) \rightarrow tt$, $1 \leqslant i \leqslant k_p$, defining each $p \in \Omega_\Pi$; furthermore, $\mathcal{R}_\Omega$ satisfies conditions (i)–(ii) in Lemma 1; (2) $\mathcal{R}_{\Omega_c} = (\Omega_c, B_{\Omega_c}, R_{\Omega_c})$ is OS-compact, its finite sorts (if any) are different from *Pred*, and is the constructor decomposition of $(\Delta \uplus \Omega_c, B_\Delta \uplus B_{\Omega_c}, R_\Delta \uplus R_{\Omega_c})$; and (3) each $p \in \Omega_\Pi$ has an associated set of *negative constrained patterns* of

the form:

$$\bigwedge_{1 \leqslant l \leqslant n_j} w^j{}_l \neq w'^j{}_l \Rightarrow p(v^j{}_1, \ldots, v^j{}_n) \neq tt, \;\; 1 \leqslant j \leqslant m_p$$

with the v^j_i, $w^j{}_l$ and $w'^j{}_l$ Ω_c-terms with variables in $Y_j = vars(p(v^j{}_1, \ldots, v^j{}_n))$. These negative constrained patterns are interpreted as meaning that the following *semantic equivalences* are valid in $\mathcal{C}_\mathcal{R}$ for each $p \in \Omega_\Pi$, where $\rho_j \in \{\rho \in [Y_j \rightarrow T_{\Omega_c}] \mid \rho = \rho!_{R,B}\}$, $B = B_\Delta \uplus B_{\Omega_c}$, and $R = R_\Delta \uplus R_{\Omega_c} \uplus R_\Pi$:

$$[p(v^j{}_1, \ldots, v^j{}_n)\rho_j] \in \mathcal{C}_\mathcal{R} \Leftrightarrow \bigwedge_{1 \leqslant l \leqslant n_j} (w^j{}_l \neq w'^j{}_l)\rho_j \wedge \bigwedge_{1 \leqslant i \leqslant n} v^j{}_i\rho_j =_B (v^j{}_i\rho_j)!_{R,B}$$

$$[p(t_1, \ldots, t_n)] \in \mathcal{C}_\mathcal{R} \Leftrightarrow \exists j \exists \rho_j \; [p(t_1, \ldots, t_n)] = [p(v^j{}_1, \ldots, v^j{}_n)\rho_j] \wedge \bigwedge_{1 \leqslant l \leqslant n_j} (w^j{}_l \neq w'^j{}_l)\rho_j$$

The first equivalence means that any instance of a negative predicate pattern by a normalized ground substitution ρ_j satisfying its constraint where the predicate's arguments are normalized is itself normalized, so that $\mathcal{C}_\mathcal{R} \models p(v^j{}_1, \ldots, v^j{}_n)\rho_j \neq tt$. It can be automatically checked by computing the non-identity variants (u, α) of the pattern term $p(v^j{}_1, \ldots, v^j{}_n)$ such that $v^j{}_i\alpha =_B (v^j{}_i\alpha)!_{R,B}$, $1 \leqslant i \leqslant n$, and then checking that all associated substitutions α for such variants invalidate the pattern's constraint. The second equivalence means that $[p(t_1, \ldots, t_n)] \in C_\mathcal{R}$ iff $[p(t_1, \ldots, t_n)]$ instantiates a negative pattern satisfying its constraint. Its proof requires a case analysis showing that for each $p \in \Omega_\Pi$ each ground instance $p(x_1, \ldots x_n)\theta$, with $\theta(x_i) =_B \theta(x_i)!_{R,B}$, $1 \leqslant i \leqslant n$, is either reducible to tt or irreducible and an instance of one of the constrained patterns.

Example 5. The module *NatSetPreds* from Example 4 satisfies above assumptions (1)–(3). Indeed, (1), including conditions (i)–(ii) in Lemma 1, follows easily from its definition and that of *NatSet*, and (2) also follows easily from the definition of *NatSet* and the remarks in Example 2. This leaves us with condition (3), where the negative constrained patterns for $\Omega_\Pi = \{\_ \subseteq \_, \_ > \_, even, odd, \_{:}Nat\}$ are the following:

- $(NS, NS') \neq NS' \Rightarrow NS \subseteq NS' \neq tt$.
- $N > N + M \neq tt$
- $even(N+N+1) \neq tt$, $even(\varnothing) \neq tt$, $(N \neq NS \wedge NS \neq \varnothing) \Rightarrow even(N, NS) \neq tt$
- $odd(N + N) \neq tt$, $odd(\varnothing) \neq tt$, $(N \neq NS \wedge NS \neq \varnothing) \Rightarrow odd(N, NS) \neq tt$
- $\varnothing{:}Nat \neq tt$, $(N \neq NS \wedge NS \neq \varnothing) \Rightarrow (N, NS){:}Nat \neq tt$.

where N and M have sort *Nat*, and NS and NS' sort *NatSet*. The first equivalence can be automatically checked as explained above. For example, the non-identity variants of $NS \subseteq NS'$ are $(tt, \{NS \mapsto \varnothing\})$, $(tt, \{NS \mapsto NS'\})$, and $(tt, \{NS' \mapsto NS, NS''\})$. Their substitutions all leave the instances of NS and NS' irreducible, and violate the constraint $(NS, NS') \neq NS'$. The second equivalence is proved in Appendix A.

The Inductive Satisfiability Decision Procedure. Assume $\mathcal{R}$ satisfies conditions (1)–(3) above and let $\Sigma = \Delta \uplus \Omega_c \uplus \Omega_\Pi$, and E be the axioms B plus the equations associated with the rules R in $\mathcal{R}$. Given a QF Σ-formula φ the procedure decides if φ is satisfiable in $\mathcal{C}_\mathcal{R}$. We can reduce the inductive validity decision problem of whether $\mathcal{C}_\mathcal{R} \models \varphi$ to deciding whether $\neg\varphi$ is unsatisfiable in $\mathcal{C}_\mathcal{R}$. Since any QF Σ-formula φ can be put in disjunctive normal form, a disjunction is satisfiable in $\mathcal{C}_\mathcal{R}$ iff one of the disjuncts is, and all predicates have been turned into functions of sort *Pred*, it is enough to decide the satisfiability of a conjunction of Σ-literals of the form $\bigwedge G \wedge \bigwedge D$, where the G are equations and the D are disequations. The procedure performs the following steps:

1. **Unification**. Satisfiability of the conjunction $\bigwedge G \wedge \bigwedge D$ is replaced by satisfiability for some conjunction in the set $\{(\bigwedge D\alpha)!_{R,B} \mid \alpha \in \mathit{VarUnif}_E(\bigwedge G)\}$, discarding any obviously unsatisfiable $(\bigwedge D\alpha)!_{R,B}$ in such a set.
2. Π-**Elimination**. After Step (1), each conjunction is a conjunction of disequalities $\bigwedge D'$. If $\bigwedge D'$ is a $\Delta \uplus \Omega_c$-formula, we go directly to Step (3); otherwise $\bigwedge D'$ has the form $\bigwedge D' = \bigwedge D_1 \wedge p(t_1, \ldots, t_n) \neq tt \wedge \bigwedge D_2$, where $p \in \Omega_\Pi$ and D_1 and/or D_2 may be empty conjunctions. We then replace $\bigwedge D'$ by all not obviously unsatisfiable conjunctions of the form:

$$(\bigwedge D_1 \wedge \bigwedge_{1 \leqslant l \leqslant n_j} {w^j}_l \neq {w'^j}_l \wedge \bigwedge D_2)\theta\alpha$$

   where $1 \leqslant j \leqslant m_p$, $W = \mathit{vars}(\bigwedge D')$, $(p(t'_1, \ldots, t'_n), \theta) \in [\![p(t_1, \ldots, t_n)]\!]^{W,\Omega}_{R,B}$, and α is a *disjoint* B_{Ω_c}-unifier of the equation $p(t'_1, \ldots, t'_n) = p({v^j}_1, \ldots, {v^j}_n)$ (i.e., sides are renamed to *share no variables* and $\mathit{ran}(\alpha) \cap (W \cup \mathit{ran}(\theta)) = \varnothing$). We use the negative constrained patterns of p and the constructor variants of $p(t_1, \ldots, t_n)$ to *eliminate* the disequality $p(t_1, \ldots, t_n) \neq tt$. If for some $p' \in \Omega_\Pi$ some disequality remains in $(\bigwedge D_1 \wedge \bigwedge D_2)\theta\alpha$, we iterate Step 2.
3. **Computation of $\Omega_c^\wedge$-Variants and Elimination of Finite Sorts**. For $\bigwedge D'$ a $\Delta \uplus \Omega_c$-conjunction of disequalities, viewed as a $(\Delta \uplus \Omega_c)^\wedge$-term its constructor $\Omega_c^\wedge$-variants are of the form $(\bigwedge D'', \gamma)$, with $\bigwedge D''$ an Ω_c-conjunction of disequalities. The variables of $\bigwedge D''$ are then $Y_{\mathit{fin}} \uplus Y_\infty$, with Y_{fin} the variables whose sorts are finite, and Y_∞ the variables with infinite sorts. Compute all normalized ground substitution τ of the variables Y_{fin} obtained by: (i) independently choosing for each variable $y \in Y_{\mathit{fin}}$ a canonical representative for the sort of y in all possible ways, and (ii) checking that for the τ so chosen $\bigwedge D''\tau$ is normalized, keeping τ if this holds and discarding it otherwise. Then $\bigwedge D'$ is satisfiable in $\mathcal{C}_\mathcal{R}$ iff some $\bigwedge D''\tau$ so obtained is B_{Ω_c}-consistent for some $\Omega_c^\wedge$-variant $(\bigwedge D'', \gamma)$ of $\bigwedge D'$.

Example 6. We can illustrate the use of the above decision procedure by proving the validity of the QF formula $\mathit{odd}(N) = tt \Leftrightarrow \mathit{even}(N) \neq tt$ in the initial algebra $\mathcal{C}_\mathcal{R}$ of *NatSetPreds*. That is, we need to show that its negation $(\mathit{odd}(N) = tt \wedge \mathit{even}(N) = tt) \vee (\mathit{odd}(N) \neq tt \wedge \mathit{even}(N) \neq tt)$ is unsatisfiable in $\mathcal{C}_\mathcal{R}$. Applying the **Unification** step to the first disjunct $\mathit{odd}(N) = tt \wedge \mathit{even}(N) = tt$

no variant unifiers are found, making this disjunct unsatisfiable. Applying the Π-**Elimination** step to the first disequality in the second disjunct $odd(N) \neq tt \wedge even(N) \neq tt$, since the only constructor variant of $odd(N)$ different from tt is the identity variant, and the only disjoint B_{Ω_c}-unifier of $odd(N)$ with the negative patterns for odd is $\{N \mapsto M + M\}$ for the (renamed) unconstrained negative pattern $odd(M + M) \neq tt$, we get the disequality $even(M + M) \neq tt$, whose normal form $tt \neq tt$ is unsatisfiable.

Theorem 3. *For FVP* $\mathcal{R} = (\Delta \uplus \Omega_c \uplus \Omega_\Pi, B_\Delta \uplus B_{\Omega_c}, R_\Delta \uplus R_{\Omega_c} \uplus R_\Pi)$ *protecting* $\mathcal{R}_\Omega = (\Omega_c \uplus \Omega_\Pi, B_{\Omega_c}, R_{\Omega_c} \uplus R_\Pi)$ *and satisfying above conditions (1)–(3), the above procedure correctly decides the satisfiability of a QF* Σ*-formula* φ *in the canonical term algebra* $\mathcal{C}_\mathcal{R}$*.*

Sort Predicates for Recursive Data Structures. Theorem 3 can be used to add sort predicates to (non-circular) recursive data structures, which can be axiomatized as the elements of an initial algebra T_Ω on a many-sorted signature of free constructors Ω. For example, lists can be so axiomatized with Ω consisting of just two sorts, *Elt*, viewed as a parametric sort of list elements, and *List*, a constant *nil* of sort *List*, and a "cons" constructor $\_;\_ : Elt\ List \to List$.

In general, however, adding to such data structures defined functions corresponding to "selectors" that can extract the constituent parts of each data structure cannot be done in a satisfactory way if we remain within a many-sorted setting. For example, for lists we would like to have selectors *head* and *tail* (the usual *car* and *cdr* in Lisp notation). For *head* the natural equation is $head(x; l) = x$. Likewise, the natural equation for *tail* is $tail(x; l) = l$. But this leaves open the problem of how to define $head(nil)$, for which no satisfactory solution exists. J. Meseguer and J.A. Goguen proposed a simple solution to this "constructor-selector" problem using initial order-sorted algebras in [25]. The key idea is the following. For each non-constant constructor symbol, say $c : A_1 \dots A_n \to B$, $n \geqslant 1$, we introduce a subsort $B_c < B$ and give the tighter typing $c : A_1 \dots A_n \to B_c$. The selector problem is now easily solved by associating to each non-constant constructor c selector functions $sel_i^c : B_c \to A_i$, $1 \leqslant i \leqslant n$, defined by the equations $sel_i^c(c(x_1, \dots, x_n)) = x_i$, $1 \leqslant i \leqslant n$. Outside the subsort B_c the selectors sel_i^c are actually undefined. For the above example of lists this just means adding a subsort $List_{\_;\_} < List$, where $List_{\_;\_}$ is usually written as *NeList* (non-empty lists), and tightening the typing of "cons" to $\_;\_ : Elt\ List \to NeList$. In this way the *head* and *tail* selectors have typings $head : NeList \to Elt$ and $tail : NeList \to List$, again with equations $head(x; l) = x$ and $tail(x; l) = l$, with x of sort *Elt* and l of sort *List*.

We have just described a general theory transformation $\Omega \mapsto (\widetilde{\Omega} \uplus \Delta, E_\Delta)$ from any MS signature Ω to an OS theory with selectors Δ. Due to space limitations, the following key facts are discussed in detail in Section 4.2 of [16]: (1) $(\widetilde{\Omega} \uplus \Delta, \varnothing, R(E_\Delta))$ is FVP with $(\widetilde{\Omega}, \varnothing, \varnothing)$ as its constructor decomposition. (2) To increase expressiveness, we can define for each subsort B_c associated with a constructor c a corresponding equationally-defined sort predicate $\_ : B_c$, thus obtaining a decomposition $(\widetilde{\Omega} \uplus \Pi \uplus \Delta, \varnothing, R(E_\Delta) \uplus R(E_\Pi))$ that is also FVP. (3)

Each sort predicate $\_:B_c$ has an associated set of *negative patterns*, so that our variant satisfiability algorithm makes satisfiability of QF formulas in the initial algebra $T_{\tilde{\Omega} \uplus \Pi \uplus \Delta / E_\Delta \uplus E_\Pi}$ *decidable*.

Example 7. (Lists of Naturals with Sort Predicates). We can instantiate the above order-sorted theory of lists with selectors *head* and *tail* by instantiating the parameter sort *Elt* to a sort *Nat* with constant 0, subsort $NzNat < Nat$, and unary constructor $s : Nat \to NzNat$ with selector $p : NzNat \to Nat$ satisfying the equation $p(s(n)) = n$. We then extend this specification with sort predicates $\_ : NzNat : Nat \to Pred$ and $\_ : NeList : List \to Pred$, defined by equations $n' : NzNat = tt$ and $l' : NeList = tt$, with n' of sort *NzNat* and l' of sort *NeList*. Their corresponding negative patterns are: $0 : NzNat \doteq tt$ and $nil : NeList \doteq tt$.

One advantage of adding these sort predicates is that some properties not expressible as QF formulas become QF-expressible. For example, to state that every number is either 0 or a non-zero number (resp. every list is either *nil* or a non-empty list) we need the formula $n = 0 \vee (\exists n')\ n = n'$ (resp. $l = nil \vee (\exists l')\ l = l'$), where n has sort *Nat* and n' sort *NzNat* (resp. l has sort *List* and l' sort *NeList*). But with sort predicates this can be expressed by means of the QF formula $n = 0 \vee n : NzNat = tt$ (resp. $l = nil \vee l : NeList = tt$).

5 Implementation

We have implemented the variant satisfiability decision procedure of Section 4 in a new prototype tool (see `http://users.dsic.upv.es/~rgutierrez/var-pred/`). The implementation consists of 11 new Maude modules (from 17 in total), 2345 new lines of code, and uses the Maude's `META-LEVEL` to carry out the procedure in a reflective way. The three steps of the variant satisfiability procedure are implemented using Maude's `META-LEVEL` functions. Let us illustrate them for *NatSetPreds*.

Example 8. We can prove the inductive validity of the formula `N - M = 0` $\Leftrightarrow$ (`M > N = tt` $\vee$ `N = M`), where `N - M` denotes `N` "monus" `M`, by showing that each conjunction in its negation, (`N - M = 0` $\wedge$ `M > N` $\neq$ `tt` $\wedge$ `N` $\neq$ `M`) $\vee$ (`N - M` $\neq$ `0` $\wedge$ `M > N = tt`) $\vee$ (`N - M` $\neq$ `0` $\wedge$ `N = M`) is unsatisfiable. For the first conjunct the algorithm's three steps are as follows. After the **unification** step, we obtain `(V2 + V3) > V2` $\neq$ `tt` $\wedge$ `V2` $\neq$ `V2 + V3`, where `V2` and `V3` are variables of sort `Natural`. Applying the Π-**elimination** step, we obtain: `V4` $\neq$ `V4 + 0`, where `V4` is a variable of sort `Natural`. After normalization, the formula becomes B_{Ω_c}-inconsistent and therefore unsatisfiable. The other two conjuncts are likewise unsatisfiable.

For a more detailed discussion of the implementation see Section 5 of [16].

6 Related Work and Conclusions

The original paper proposing the concepts of variant and FVP is [9]. FVP ideas have been further advanced in [13,7,4,6]. Variant satisfiability has been studied

in [22,21,28]. In relation to that work, the main contribution of this paper is the extension of variant satisfiability to handle user-definable predicates.

As mentioned in the Introduction, satisfiability decision procedures can be either theory-specific or theory-generic. These two classes of procedures complement each other: theory specific ones are more efficient; but theory-generic ones are user-definable and can substantially increase the range of SMT solvers. On theory-specific decision procedures advanced textbooks include, e.g., [5,18], and work on data type satisfiability includes, e.g., [29,3,11]. In relation to theory-specific work, what the results in this paper provide is a *generic algorithm* for a wide class of user-definable data types *with user-definable predicates*.

Other theory-generic satisfiability approaches include: (i) the superposition-based one, e.g., [19,2,20,1,30], where it is proved that a superposition theorem proving inference system terminates for a given first-order theory together with any given set of ground clauses representing a satisfiability problem; and (ii) that of decidable theories defined by means of formulas with triggers [12], that allows 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. While not directly comparable to the present one, these approaches (discussed in considerably greater detail in [22,21]) can be seen as complementary ones, further enlarging the repertoire of theory-generic satisfiability methods.

In conclusion, the present work has extended variant satisfiability to support initial algebras specified by FVP theories with user-definable predicates under fairly general conditions. Since such predicates are often needed in specifications, this substantially enlarges the scope of variant-based initial satisfiability algorithms. The most obvious next step is to combine the original variant satisfiability algorithm defined in [22,21] and implemented in [28] with the present one. To simplify both the exposition and the prototype implementation, a few simplifying assumptions, such as the assumption that the signature Ω of constructors and that Δ of defined functions share no subsort-overloaded symbols, have been made. For both greater efficiency and wider applicability, the combined generic algorithm will drop such assumptions and will use constructor unification [22,28].

References

1. Armando, A., Bonacina, M.P., Ranise, S., Schulz, S.: New results on rewrite-based satisfiability procedures. ACM TCL. 10(1) (2009)
2. Armando, A., Ranise, S., Rusinowitch, M.: A rewriting approach to satisfiability procedures. I&C 183(2), 140–164 (2003)
3. Barrett, C., Shikanian, I., Tinelli, C.: An abstract decision procedure for satisfiability in the theory of inductive data types. JSAT 3, 21–46 (2007)
4. 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)
5. Bradley, A.R., Manna, Z.: The calculus of computation - decision procedures with applications to verification. Springer (2007)

6. Cholewa, A., Meseguer, J., Escobar, S.: Variants of variants and the finite variant property. Tech. rep., http://hdl.handle.net/2142/47117
7. Ciobaca., S.: Verification of Composition of Security Protocols with Applications to Electronic Voting. Ph.D. thesis, ENS Cachan (2011)
8. Comon, H.: Complete axiomatizations of some quotient term algebras. TCS 118(2), 167–191 (1993)
9. 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
10. 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)
11. 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)
12. Dross, C., Conchon, S., Kanig, J., Paskevich, A.: Adding decision procedures to SMT solvers using axioms with triggers. JAR 56(4), 387–457 (2016)
13. Escobar, S., Sasse, R., Meseguer, J.: Folding variant narrowing and optimal variant termination. JALP 81, 898–928 (2012)
14. Goguen, J., Meseguer, J.: Models and equality for logical programming. In: Proc. TAPSOFT'87, Springer LNCS, vol. 250, pp. 1–22. Springer (1987)
15. Goguen, J., Meseguer, J.: Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. TCS 105, 217–273 (1992)
16. Gutiérrez, R., Meseguer, J.: Variant-based decidable satisfiability in initial algebras with predicates. Tech. Rep. http://hdl.handle.net/2142/96264, University of Illinois at Urbana-Champaign (June 2017)
17. Jouannaud, J.P., Kirchner, H.: Completion of a set of rules modulo a set of equations. SIAM J. of Computing 15, 1155–1194 (November 1986)
18. Kroening, D., Strichman, O.: Decision Procedures - An Algorithmic Point of View. Texts in Theoretical Computer Science. An EATCS Series, Springer (2008)
19. Lynch, C., Morawska, B.: Automatic decidability. In: Proc. LICS 2002. p. 7. IEEE Computer Society (2002)
20. Lynch, C., Tran, D.: Automatic decidability and combinability revisited. In: Proc. CADE 2007. vol. 4603, pp. 328–344. Springer LNCS (2007)
21. Meseguer, J.: Variant-based satisfiability in initial algebras. In: Artho, C., Ölveczky, P. (eds.) Proc. FTSCS 2015. pp. 1–32. Springer CCIS 596 (2016)
22. Meseguer, J.: Variant-based satisfiability in initial algebras. Sci. Comp. Prog. (2017), to appear; 2015 Tech. Rep. http://hdl.handle.net/2142/88408
23. Meseguer, J.: Strict coherence of conditional rewriting modulo axioms. TCS 672, 1–35 (2017)
24. Meseguer, J., Goguen, J.: Initiality, induction and computability. In: Nivat, M., Reynolds, Algebraic Methods in Semantics, pp. 459–541. Cambridge (1985)
25. Meseguer, J., Goguen, J.: Order-sorted algebra solves the constructor-selector, multiple representation and coercion problems. Info.&Comp. 103(1), 114–158 (1993)
26. Nelson, G., Oppen, D.C.: Simplification by cooperating decision procedures. ACM TOPLAS 1(2), 245–257 (1979)
27. Shostak, R.E.: Deciding combinations of theories. J. of the ACM 31(1), 1–12 (1984)
28. Skeirik, S., Meseguer, J.: Metalevel algorithms for variant-based satisfiability. In: Lucanu, D. (ed.) Proc. WRLA 2016. vol. 9942, pp. 167–184. Springer LNCS (2016)
29. 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 (2001)
30. Tushkanova, E., Giorgetti, A., Ringeissen, C., Kouchnarenko, O.: A rule-based system for automatic decidability and combinability. SCP 99, 3–23 (2015)

A Sufficient Completeness of Patterns for *NatSetPreds*

We need to prove that for all predicates in *NatSetPreds* the positive patterns given by the defining rules and the negative constrained patterns cover all ground instances and are therefore "sufficiently complete" in the precise sense that for each $p \in \Omega_\Pi$ in *NatSetPreds* each ground instance $p(x_1, \ldots x_n)\theta$, with $\theta(x_i) =_B \theta(x_i)!_{R,B}$, $1 \leqslant i \leqslant n$, is either reducible to tt or irreducible and an instance of one of the above constrained patterns.

For $\subseteq$, as shown in Example 5, all non-identity variants (tt, α) of the term $NS \subseteq NS'$ are such that α E-unifies $(NS, NS') = NS'$ and therefore violates the constraint $(NS, NS') \neq NS'$. This proves that $\mathcal{C}_\mathcal{R} \models NS \subseteq NS' = tt \Rightarrow (NS, NS') = NS'$ holds. We will be done if we prove $\mathcal{C}_\mathcal{R} \models NS \subseteq NS' = tt \Leftarrow (NS, NS') = NS'$, since then $\mathcal{C}_\mathcal{R} \models NS \subseteq NS' \neq tt \Leftrightarrow (NS, NS') \neq NS'$. But this trivially holds by application of the rule $NS \subseteq NS, NS' \rightarrow tt$.

For the $\_ > \_$ predicate we need to show that the positive pattern $N+M+1 > N$ and the negative pattern $N > N + M$ cover all ground instances of $\_ > \_$ by irreducible ground arguments. The key observations are that: (i) a natural number in this representation is a (possibly empty) multiset of 1's, (ii) given any two such multisets J and K, one of the two mutually exclusive alternatives, $J \supset K$ or $J \subseteq K$, holds, (iii) $J \supset K$ holds iff $(\exists M)\ J = M+K+1$ iff $J > K = tt$, and (iv) $J \subseteq K$ holds iff $(\exists M)\ K = M + K$ iff $J > K \neq tt$.

The $\_ : Nat$ predicate, has the positive pattern $N : Nat$ with N of sort Nat and the negative patterns $\varnothing : Nat \neq tt$, $(N \neq NS \wedge NS \neq \varnothing) \Rightarrow (N, NS) : Nat \neq tt$. Note that any irreducible, non-singleton finite set of natural numbers is either empty, so that the first negative pattern applies, or is non-empty and has the form $n_1, \ldots, n_k$, $k > 1$, with $n_1, \ldots, n_k$ natural numbers and $n_i \neq n_j$ if $1 \leqslant i < j \leqslant k$. Then observe that any match of $n_1, \ldots, n_k$ modulo associativity and commutativity with the pattern N, NS (corresponding to choosing some n_i as the instance of N) satisfies the second pattern's constraint.

For the *even* and *odd* predicates, since both cases are entirely similar, it is enough to show sufficient completeness for the case of the *even* predicate. Since any ground term $even(u)$ reducible by the rule for *even* (with u irreducible) must have sort *Nat*, if u does not have sort *Nat*, the same argument given for the $\_ : Nat$ predicate shows that it must be an instance of either $even(\varnothing) \neq tt$ or $(N \neq NS \wedge NS \neq \varnothing) \Rightarrow even(N, NS) \neq tt$. So we only need to show that the positive pattern $even(N + N)$ and the negative one $even(M + M + 1)$ cover all natural numbers. But this follows from the inductive theorem: $(\forall x)\ ((\exists n)\ x = n + n \vee (\exists m)\ x = m + m + 1)$, which has an easy proof by induction on x.