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 those the Nelson and Oppen [29], or Shostak [30], T_0 must be a finite combination of the decidable theories $Q_1, \ldots, Q_k$ supported by the SMT solver.

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 sigle 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 [24,25] 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 [16] signature of function symbols, supporting types, subtypes, and subtype polymorphism; (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* [24,25]. 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 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 constructors $T_{\Omega/E_\Omega\cup B_\Omega}$. 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 [24,25], 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 $p : A_1, \ldots, A_n \rightarrow \mathit{Pred}$, 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.

Section 2 gathers preliminaries. Constructor variants and OS-compactness are discussed in Section 3. The theory-generic 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.

2 Order-Sorted Algebra, Rewriting, and Variants

We summarize the order-sorted algebra, order-sorted rewriting, and finite variant notions needed in the paper. The material, adapted from [26,14]. It assumes the notions of many-sorted signature and many-sorted algebra, e.g., [13], 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 of sorts and* (S, Σ) *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* $s'_i \leqslant s_i$*,* $1 \leqslant i \leqslant n$*.* Σ *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 Σ *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 $[s_1], \ldots, [s_n], [s] \in \widehat{S}$

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

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

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

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

An order-sorted Σ-homomorphism $h : A \to B$ *is a many-sorted* (S, Σ)*-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* Σ*-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. *[26] The category* $\mathbf{OSAlg}_\Sigma$ *has an initial algebra. Furthermore, if* Σ *is sensible, then the term algebra* T_Σ *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* $f : s_1 \ldots s_n \to s$ *and* $t_i \in T_{\Sigma,s_i}$ $1 \leqslant i \leqslant n$*, then* $f(t_1, \ldots, t_n) \in T_{\Sigma,s}$*,*

is initial, i.e., there is a unique Σ*-homomorphism from* T_Σ *to each* Σ*-algebra.*

T_Σ 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 Σ is said to *have non-empty sorts* iff for each $s \in S$, $T_{\Sigma,s} \neq \emptyset$. *We will assume throughout that Σ has non-empty sorts.* An OS signature Σ is called *preregular* [16] iff for each $t \in T_\Sigma$ the set $\{s \in S \mid t \in T_{\Sigma,s}\}$ has a least element, denoted $ls(t)$. *We will assume throughout that Σ 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 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 $T_\Sigma(X)$ can be expressed as the following corollary of Theorem 1:

Theorem 2. *(Freeness Theorem). If Σ is sensible, for each $A \in \mathbf{OSAlg}_\Sigma$ 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 $A = T_\Sigma(X)$, 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 ρ that just renames a few variables and may lower their sort. More precisely, $dom(\rho)$ is a finite set of variables $\{x_1, \ldots, x_n\}$, with respective sorts $s_1, \ldots, s_n$, and ρ injectively maps the $x_1, \ldots, x_n$ to variables $x'_1, \ldots, x'_n$ with respective sorts $s'_1, \ldots, s'_n$ such that $s'_i \leqslant s_i$, $1 \leqslant i \leqslant n$.

The first-order language of *equational Σ-formulas* is defined in the usual way: its atoms are *Σ-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 Σ-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 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* $A, \alpha \models \varphi$ is defined inductively as usual. We say that φ is *valid* in A, denoted $A \models \varphi$, iff $A, \alpha \models \varphi$ holds for each $\alpha \in [Y \to A]$, where $Y = fvars(\varphi)$. We say that φ is *satisfiable* in A iff $\exists \alpha \in [Y \to A]$ such that $A, \alpha \models \varphi$, where $Y = fvars(\varphi)$. 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 Ω and discards everything in $\Sigma - \Omega$. If $\varphi \in Form(\Omega)$ 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 Σ-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 *(Σ, E)-algebras*. $\mathbf{OSAlg}_{(\Sigma,E)}$ has an *initial algebra* $T_{\Sigma/E}$ [26]. The inference system in [26] is *sound and complete* for OS equational deduction, i.e., for any OS equational theory (Σ, E), and Σ-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 Σ is called *E-preregular* iff for each $u = v \in E$ and variable specialization ρ, $ls(u\rho) = ls(v\rho)$.

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 $p : s_1 \ldots s_n \to Pred$, with $Pred$, a new sort having a constant tt. The reduction to OS algebra is achieved as follows. An OS-FO signature, is a pair (Σ, Π) with Σ an OS signature with set of sorts S, and Π an S^*-indexed set $\Pi = \{\Pi_w\}_{w \in S^*}$ of *predicate symbols*. We associate to an OS-FO signature (Σ, Π) an OS signature $(\Sigma \cup \Pi)$ by adding to Σ a new sort $Pred$ with a constant tt in its own separate connected component $\{Pred\}$, and viewing each $p \in \Pi_w$ as a function symbol $p : s_1 \ldots s_n \to Pred$. The reduction at the model level is now very simple: each OS $(\Sigma \cup \Pi)$-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 $p(t_1, \ldots, t_n)$ 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 (Σ, Π) an OS-FO signature and Γ a set of (Σ, Π)-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 $T_{(\Sigma,\Pi),\Gamma}$ [15]. A useful, easy to check fact is that we have an identity: $T^\circ_{\Sigma \cup \Pi/\widetilde{\Gamma}} = T_{(\Sigma,\Pi),\Gamma}$. We will give various natural examples of OS-FO equational theories (in the form $(\Sigma \cup \Pi, \widetilde{\Gamma})$) later in the paper.

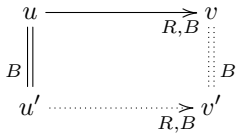
Recall the notation for term positions, subterms, and term replacement from [10]: (i) positions in a term viewed as a tree are marked by strings $p \in \mathbb{N}^*$ specifying a path from the root, (ii) $t|_p$ denotes the subterm of term t at position p, and (iii) $t[u]_p$ denotes the result of *replacing* subterm $t|_p$ at position p by u.

Definition 3. *A* rewrite theory *is a triple* $\mathcal{R} = (\Sigma, B, R)$ *with* (Σ, B) *an order-sorted 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* $[s] \in \widehat{S}$*. 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* ρ, $ls(l\rho) \geqslant ls(r\rho)$.
3. Σ *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* $t \rightarrow_{R,B} t'$*, holds between* $t, t' \in T_\Sigma(X)_{[s]}$*,* $[s] \in \widehat{S}$*, iff there is a rewrite rule* $l \rightarrow r \in R$*, a substitution* $\sigma \in [X \rightarrow T_\Sigma(X)]$*, 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* Σ*-term.*

$\mathcal{R}$ *is called: (i)* terminating *iff the relation* $\rightarrow_{R,B}$ *is well-founded; (ii)* strictly B-coherent *[27] iff whenever* $u \rightarrow_{R,B} v$ *and* $u =_B u'$ *there is a* v' *such that* $u' \rightarrow_{R,B} v'$ *and* $v =_B v'$:



(iii) confluent *iff* $u \rightarrow^*_{R,B} v_1$ *and* $u \rightarrow^*_{R,B} v_2$ *imply that there are* w_1, w_2 *such that* $v_1 \rightarrow^*_{R,B} w_1$*,* $v_2 \rightarrow^*_{R,B} w_2$*, and* $w_1 =_B w_2$ *(where* $\rightarrow^*_{R,B}$ *denotes the reflexive-transitive closure of* $\rightarrow_{R,B}$*); and (iv)* convergent *if (i)–(iii) hold. If* $\mathcal{R}$ *is convergent, for each* Σ*-term* t *there is a term* u *such that* $t \rightarrow^*_{R,B} u$ *and* $(\nexists v)\ u \rightarrow_{R,B} v$*. 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 Σ-equations, let $R(E) = \{u \rightarrow v \mid u = v \in E\}$. A *decomposition* of an order-sorted 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 3. *(Church-Rosser Theorem) [20,27] 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 $f : s_1 \dots s_n \rightarrow s$ 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 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. To simplify the exposition we assume throughout that for each subsort-polymorphic family of function symbols $f_{[s]}^{[s_1]\cdots[s_n]}$ either $f_{[s]}^{[s_1]\cdots[s_n]} \subseteq \Omega$ or $f_{[s]}^{[s_1]\cdots[s_n]} \subseteq \Delta$. Tools based on tree automata [7], equational tree automata [19], or narrowing [17], can be used to automatically check sufficient completeness of a decomposition $\mathcal{R}$ with respect to constructors Ω under some assumptions.

As the following definition shows, sufficient completeness is closely related to the notion of a *protecting* theory inclusion.

Definition 4. *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 an OS 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 5. *Given a decomposition* $\mathcal{R} = (\Sigma, B, R)$ *of an OS equational theory* (Σ, E) *and a* Σ*-term* t, *a* variant[3] *[9,14] 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 [5]. Here we follow the shaper formulation in [14], rather than the one in [9], because it is technically essential for some results to hold [5].

$(t\theta)!_{R,B}$, *(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* $[\![t]\!]_{R,B} = \{(u_1, \theta_1), \ldots, (u_n, \theta_n)\}$. *Since if* B *has a finitary* B*-unification algorithm the relation* $(u, \alpha) \sqsupseteq_B (v, \beta)$ *is decidable by* B*-matching, in that case we can always assume that 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* $ran(\theta_i) \cap W = \varnothing$. *As for unifiers,* $[\![t]\!]^{exp_1,\ldots,exp_n}_{R,B}$ *abbreviates* $[\![t]\!]^W_{R,B}$, *where* $W = vars(\phi) \cup \bigcup_{1 \leqslant i \leqslant n} vars(exp_i)$.

If B has a finitary unification algorithm, the *folding variant narrowing* strategy described in provides an effective method to generate $[\![t]\!]_{R,B}$ [14]. Furthermore, folding variant narrowing *terminates* for each input $t \in T_\Sigma(X)$ with a finite set $[\![t]\!]_{R,B}$ iff $\mathcal{R}$ is FVP [14].

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*, *NatSet* and *Pred*, subsort inclusion $Nat < NatSet$, 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*, $\_,\_ : NatSet\ NatSet \to NatSet$ (set union), *tt* of sort *Pred*, and a subset containment predicate expressed as a function $\_ \subseteq \_ : NatSet\ NatSet \to Pred$. 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 $NS, \varnothing \to NS$; (ii) idempotency rules for union $NS, NS \to NS$, and $NS, NS, NS' \to NS, NS'$; and (iii) rules defining the $\_ \subseteq \_$ predicate, $\varnothing \subseteq NS \to tt$, $NS \subseteq NS \to tt$, and $NS \subseteq NS, NS' \to tt$, where NS and NS' have sort *NatSet*. 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" 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*.

The predicates $\in$ and $\subset$ need not be explicitly defined, since they can be expressed by the definitional equivalences $N \in NS = tt \Leftrightarrow N, NS = NS$, and $NS \subset NS' = tt \Leftrightarrow NS \subseteq NS' = tt \ \wedge \ NS \neq NS'$.

FVP is a *semi-decidable* property [5], which can be easily verified (when it holds) by checking, using folding variant narrowing (supported by Maude 2.7), 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 *NatSet* above is 20.

Folding variant narrowing provides a method for generating a *complete set of E-unifiers*. Let (Σ, E) have a decomposition $\mathcal{R} = (\Sigma, B, R)$ with B having a finitary B-unification algorithm, which we assume extensible by the addition of free function symbols to Σ. 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 Σ 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 $\_ \neq \_ : \top_{[s]}\ \top_{[s]} \to Lit$.

Variant-based unification goes back to [14]. The following theorem, proved in detail in [24], gives a more precise characterization using $\Sigma^\wedge$-terms.

Theorem 4. *Under the above assumptions on $\mathcal{R}$, let ϕ be a system of Σ-equations viewed as a $\Sigma^\wedge$-term of sort Conj. Then, for any finite set W of variables $W \supseteq vars(\phi)$, the set $VarUnif_E^W(\phi)$ of* variant E-unifiers *of ϕ away from W is by definition the set:*

$$\{(\theta\gamma)|_{vars(\phi)} \mid (\phi', \theta) \in [\![\phi]\!]^W_{R,B} \wedge \gamma \in Unif_B^{W,\theta}(\phi') \wedge (\phi'\gamma, (\theta\gamma)|_{vars(\phi)})\ R,B \text{ variant of } \phi\}$$

$VarUnif_E^W(\phi)$ is a complete *set of E-unifiers of ϕ away from W in the strong sense that if α is an R,B-normalized E-unifier of ϕ there exists $(\theta\gamma)|_{vars(\phi)} \in VarUnif_E^W(\phi)$, were $(\phi', \theta) \in [\![\phi]\!]^W_{R,B}$, and an R,B-*normalized *ρ such that: (i) $\alpha|_W =_B ((\theta\gamma)|_{vars(\phi)}\rho)|_W$; and (ii) $(\phi\alpha)!_{R,B} =_B \phi'\gamma\rho$.*

Furthermore, when $\mathcal{R}$ is FVP, the generation of $VarUnif_E^W(\phi)$ by folding variant narrowing of ϕ followed by B-unification always terminates *with a* finite *set of unifiers, thus providing a finitary E-unification algorithm.*

3 Constructor Variants and OS-Compactness

We gather some technical notions and results needed for the inductive satisfiability procedure given in Section 4. The results on constructor variants are new.

They complement previous such results in [31] for free constructors modulo axioms and in [24] for many-sorted constructor variants.

The notion of *constructor variant* answers the question: what variants of t cover as instances modulo B_Ω all canonical forms of all ground instances of t? The following lemma gives a precise answer under reasonable assumptions:

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) $\Sigma = \Omega \uplus \Delta$ and for each subsort-polymorphic family of operators $f^{[s_1]\ldots[s_n]}_{[s]}$ in Σ either $f^{[s_1]\ldots[s_n]}_{[s]} \subseteq \Omega$ or $f^{[s_1]\ldots[s_n]}_{[s]} \subseteq \Delta$; (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 $u =_B v\tau$.*

Proof. Suppose $[u] \in \mathcal{C}_{\mathcal{R}_\Omega}$ is of the form $u =_B (t\gamma)!_{R,B}$. We may assume without loss of generality that $\gamma = (\gamma)!_{R,B}$, so that (u,γ) is a ground variant of t. Therefore, there exists $(v,\theta) \in [\![t]\!]_{R,B}$ and a normalized substitution τ such that $u =_B v\tau$. We may assume τ ground by restricting it to the variables of v and by the axioms B being regular. We will be done if we show that $v \in T_\Omega(X)$. Suppose not. Then v must contain some symbol in Δ, and a fortiori $v\tau$ does. Since $u =_B v\tau$ there is an equality proof, i.e., a sequence of equality steps using axioms in either B_Ω or B_Δ. But since all axioms in B are regular and linear and Ω and Δ are disjoint, if $w \notin T_\Omega(X)$ and $w =_{B_\Omega} w'$, then $w' \notin T_\Omega(X)$; and by the further assumption on B_Δ, if $w =_{B_\Delta} w'$, then $w, w' \notin T_\Omega(X)$. But since $u \in T_\Omega$, this makes $u =_B v\tau$ impossible. □

We finally need the notion of an OS-*compact* equational OS-FO theory $((\Sigma, \Pi), \Gamma)$, generalizing the compactness notion in [8]. Given an OS equational theory (Σ, E), call a Σ-equality $u = v$ *E-trivial* iff $u =_E v$, and a Σ-disequality $u \neq v$ *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 6. *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 $T_{\Sigma \cup \Pi/\widetilde{\Gamma},}$, 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 (Σ, Π)-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 (Σ, E) OS-*compact *iff the OS-FO theory $((\Sigma, \varnothing), E)$ is so.*

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

Theorem 5. *[24,25] 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 [24]: (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* with set containment predicate $\_ \subseteq \_$ is just the instantiation of the constructor decomposition for the parameterized module of (finite) sets in [24] to the natural numbers with 0, 1, and $\_ + \_$, which is 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 [24,25] apply to the initial algebra $T_{\Sigma/E}$ of an equational theory (Σ, E) having an FVP variant-decomposition $\mathcal{R} = (\Sigma, B, R)$ protecting a constructor decomposition $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$ and such that: (i) B has a finitary unification algorithm; and (iii) 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 [24,25] 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 [24], and a prototype Maude implementation is presented in [31]. 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 $p : s_1, \ldots, s_n \to \mathit{Pred}$ 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, which in some cases is impossible. 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 [24] required non-trivial analyses.

Example 4. Consider the following extension by predicates *NatSetPreds* of the *NatSet* theory in Example 1, where the constructor signature $\Omega = \Omega_c \uplus \Omega_\Pi$ adds the subsignature Ω_Π containing the strict order predicate $\_ > \_ : Nat\ Nat \rightarrow Pred$, the "sort predicate" $\_ : Nat : NatSet \rightarrow Pred$, characterizing when a set of natural numbers is a natural, and the even and odd predicates $even, odd : NatSet \rightarrow Pred$, defined by the rules R_Π: $N + M + 1 > N \rightarrow tt$, $N : Nat \rightarrow tt$, $even(N + N) \rightarrow tt$, $odd(N + N + 1) \rightarrow tt$, where 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 of disequalities $N > M \neq tt \wedge M > N \neq tt \wedge N \neq M$.

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 $p(u^i_1, \ldots, u^i_n) \rightarrow tt$, $1 \leqslant i \leqslant k_p$, 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:

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

where the ${w^j}_l$ and ${w'^{,j}}_l$ are Ω_c-terms with variables among those 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 $R = R_\Delta \uplus R_{\Omega_c} \uplus R_\Pi$:

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

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

The first equivalence means that any instance of a negative pattern by a normalized ground substitution ρ_j satisfying its constrain is normalized, so that

$\mathcal{C}_{\mathcal{R}} \models p(v^j{}_1, \ldots, v^j{}_n)\rho_j \neq tt$. The second 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.

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

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

where N and M have sort *Nat* and NS sort *Natset*. As explained in Appendix A, the first equivalence can be automatically checked using variant satisfiability. That the two equivalences hold in $\mathcal{C}_{\mathcal{R}}$ for these predicates and their patterns is proved in detail in Appendix A.

4.1 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 to 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 $\{(\bigwedge D\alpha)!_{R,B} \mid \alpha \in VarUnif_E(\bigwedge G)\}$, 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 $(p(t'_1, \ldots, t'_n), \theta) \in [\![p(t_1, \ldots, t_n)]\!]^{\Omega}_{R,B}$, 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., the two sides are renamed to

share no variables), and $1 \leqslant j \leqslant m_p$. That is, 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 any such disequalities remain in $(\bigwedge D_1 \wedge \bigwedge D_2)\theta\alpha$ for some $p' \in \Omega_\Pi$, we keep applying Step 2 until none remains.

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 $Y_{fin} \uplus Y_\infty$, 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 $y \in Y_{fin}$ 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 $odd(N) = tt \Leftrightarrow even(N) \neq tt$ in the initial algebra $\mathcal{C}_\mathcal{R}$ of *NatSetPreds*. That is, we need to show that its negation $(odd(N) = tt \wedge even(N) = tt) \vee (odd(N) \neq tt \wedge even(N) \neq tt)$ is unsatisfiable in $\mathcal{C}_\mathcal{R}$. Applying the **Unification** step to the first disjunct $odd(N) = tt \wedge even(N) = tt$ 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 6. *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 $\mathcal{C}_\mathcal{R}$.*

Proof. The procedure clearly terminates. In particular, the Π-**Elimination** step reduces the number of Π-symbols in a conjunction by one. Since each step transforms a conjunction into a set of other conjunctions, we will be done if we show that in each such transformation the given conjunction and the resulting set of conjunctions (viewed as a disjunction) are equi-satisfiable in $\mathcal{C}_\mathcal{R}$.

Unification. If $(\bigwedge D\alpha)!_{R,B}$ with $\alpha \in VarUnif_E(\bigwedge G)\}$ is satisfiable in $\mathcal{C}_\mathcal{R}$ by some normalized ground substitution ρ, then $(\alpha\rho)!_{R,B}$ satisfies $\bigwedge G \wedge \bigwedge D$ in $\mathcal{C}_\mathcal{R}$. Conversely, if a normalized ground substitution ρ satisfies $\bigwedge G \wedge \bigwedge D$ in $\mathcal{C}_\mathcal{R}$, by the Church-Rosser Theorem ρ is a ground E-unifier of $\bigwedge G$ and therefore there exists $\alpha \in VarUnif_E^W(\bigwedge G)\}$ and a normalized substitution γ such that $\rho|_W =_B (\alpha\gamma)|_W$ with $W = vars(\bigwedge G \wedge \bigwedge D)$. But this means that γ is a satisfying assignment for $(\bigwedge D\alpha)!_{R,B}$ in $\mathcal{C}_\mathcal{R}$.

Π-**Elimination**. If a normalized ground substitution ρ is a satisfying assignment for $(\bigwedge D_1 \wedge \bigwedge_{1 \leqslant l \leqslant n_j} w^j{}_l \neq w'^{,j}{}_l \wedge \bigwedge D_2)\theta\alpha$ in $\mathcal{C}_{\mathcal{R}}$, where $(p(t'_1, \ldots, t'_n), \theta) \in [\![p(t_1, \ldots, t_n)]\!]^{\Omega}_{R,B}$, α is a B_{Ω_c}-unifier of the equation $p(t'_1, \ldots, t'_n) = p(v^j{}_1, \ldots, v^j{}_n)$, and and $1 \leqslant j \leqslant m_p$, then by the first semantic fact about negative constrained patterns we have that $(p(t_1, \ldots, t_n)\theta\alpha\rho)!_{R,B} = p(v^j{}_1, \ldots, v^j{}_n)\alpha\rho$, so that $(\theta\alpha\rho)!_{R,B}$ is a satisfying assignment for $\bigwedge D_1 \wedge p(t_1, \ldots, t_n) \neq tt \wedge \bigwedge D_2$ in $\mathcal{C}_{\mathcal{R}}$. Conversely, if ρ is a normalized ground substitution satisfying $\bigwedge D_1 \wedge p(t_1, \ldots, t_n) \neq tt \wedge \bigwedge D_2$ in $\mathcal{C}_{\mathcal{R}}$, then $(p(t_1, \ldots, t_n)\rho)!_{R,B}$ must be different from tt, i.e., of the form $p(u_1, \ldots, u_n)$. Let $W = vars(\bigwedge D_1 \wedge p(t_1, \ldots, t_n) \neq tt \bigwedge D_2)$ and $W_p = vars(p(t_1, \ldots, t_n))$, so that $W = W_p \uplus W'$. By the notion of constructor variant we must have $(p(t'_1, \ldots, t'_n), \theta) \in [\![p(t_1, \ldots, t_n)]\!]^{\Omega}_{R,B}$ and a normalized a ground substitution γ such that $(p(t_1, \ldots, t_n)\rho)!_{R,B} =_B p(t'_1, \ldots, t'_n)\gamma = (p(t'_1, \ldots, t'_n)\gamma)!_{R,B}$ and $\rho|_{W_p} =_B \theta\gamma$, so that $\rho|_{W'} \uplus \theta\gamma$ is also a normalized ground substitution satisfying $\bigwedge D_1 \wedge p(t_1, \ldots, t_n) \neq tt \wedge \bigwedge D_2$ in $\mathcal{C}_{\mathcal{R}}$ in $\mathcal{C}_{\mathcal{R}}$. But by the second semantic fact about negative constrained patterns we have a j, $1 \leqslant j \leqslant m_p$ and a normalized ground substitution ρ_j of the variables Y_j such that $p(t'_1, \ldots, t'_n)\gamma =_B p(v^j{}_1, \ldots, v^j{}_n)\rho_j$ and $\bigwedge_{1 \leqslant l \leqslant n_j} (w^j{}_l \neq w'^{,j}{}_l)\rho_j$ holds in $\mathcal{C}_{\mathcal{R}}$. Therefore, $\gamma \uplus \rho_j$ is a B unifier of the equation $p(t'_1, \ldots, t'_n) = p(v^j{}_1, \ldots, v^j{}_n)$ so that there is a (disjoint) B-unifier α and a normalized ground substitution δ such that $(\alpha\delta)|_{Y_j \uplus vars(p(t'_1, \ldots, t'_n))} =_B \gamma \uplus \rho_j$. Therefore, $\rho|_{W'} \uplus \delta$ is a normalized ground substitution satisfying $(\bigwedge D_1 \wedge \bigwedge_{1 \leqslant l \leqslant n_j} w^j{}_l \neq w'^{,j}{}_l \wedge \bigwedge D_2)\theta\alpha$ in $\mathcal{C}_{\mathcal{R}}$.

Computation of $\Omega_c^{\wedge}$-Variants and **Elimination of Finite Sorts**. Let $\bigwedge D'$ be a $\Delta \uplus \Omega_c$-conjunction of disequalities and ρ a normalized ground substitution satisfying $\bigwedge D'$ in $\mathcal{C}_{\mathcal{R}}$. Viewing $\bigwedge D'$ as a $(\Delta \uplus \Omega_c)^{\wedge}$-term this implies that there is a normalized ground substitution τ and a constructor $\Omega_c^{\wedge}$-variant $(\bigwedge D'', \gamma)$ of $\bigwedge D'$ such that $(\bigwedge D'\rho)|_{R,B} =_{B_{\Omega_c}} \bigwedge D''\tau$. Therefore, τ is a satisfying assignment for $\bigwedge D''$ in $\mathcal{C}_{\mathcal{R}}$. But $\bigwedge D''\tau$ normalized implies that $\bigwedge D''\tau|_{Y_{fin}}$ is normalized, with $\tau|_{Y_{fin}}$ a choice of canonical representatives for the variables Y_{fin} up to B_{Ω_c}-equality, and that $\tau|_{Y_\infty}$ is a satisfying assignment for $\bigwedge D''\tau|_{Y_{fin}}$ in $\mathcal{C}_{\mathcal{R}}$. Therefore, $\bigwedge D''\tau|_{Y_{fin}}$ normalized, the Church-Rosser property, and the fact that $\tau|_{Y_\infty}$ is a satisfying assignment for $\bigwedge D''\tau|_{Y_{fin}}$ in $\mathcal{C}_{\mathcal{R}}$ force $\bigwedge D''\tau|_{Y_{fin}}$ to be B_{Ω_c}-consistent. Conversely, let $(\bigwedge D'', \gamma)$, with variables $Y_{fin} \uplus Y_\infty$, be a constructor $\Omega_c^{\wedge}$-variant of $\bigwedge D'$ and let $\tau|_{Y_{fin}}$ be a canonical choice of representatives such that $\bigwedge D''\tau|_{Y_{fin}}$ is normalized and B_{Ω_c}-consistent. Then, by the Church-Rosser property $\bigwedge D''\tau|_{Y_{fin}}$ is also E_{Ω_c}-consistent for E_{Ω_c} the equations associated to the decomposition $\mathcal{R}_{\Omega_c} = (\Omega_c, B_{\Omega_c}, R_{\Omega_c})$. Therefore, by assumption (2) there is a normalized satisfying assignment $\tau|_{Y_\infty}$ for $\bigwedge D''\tau|_{Y_{fin}}$ in $\mathcal{C}_{\mathcal{R}}$, so that $(\gamma(\tau|_{Y_{fin}} \uplus \tau|_{Y_\infty}))!_{R,B}$ is a satisfying assignment for $\bigwedge D'$ in $\mathcal{C}_{\mathcal{R}}$. □

4.2 Sort Predicates for Recursive Data Structures

We can axiomatize many (non-circular) recursive data structures 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 $\_;\_ : Elt\ List \rightarrow List$.

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 [28]. The key idea is the following. For each non-constant constructor symbol, say $c : A_1 \dots A_n \rightarrow B$, $n \geqslant 1$, we introduce a subsort $B_c < B$ and give the tighter typing $c : A_1 \dots A_n \rightarrow B_c$. The selector problem is now easily solved by associating to each non-constant constructor c selector functions $sel_i^c : B_c \rightarrow A_i$, $1 \leqslant i \leqslant n$, 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 $\_;\_ : Elt\ List \rightarrow NeList$. In this way the *head* and *tail* selectors have typings $head : NeList \rightarrow Elt$ and $tail : NeList \rightarrow List$, again with equations $head(x;l) = x$ and $tail(x;l) = l$, with x of sort *Elt* and l of sort *List*.

The key facts not just for lists, but for any recursive data structure whose selectors Δ have been defined by means of an order-sorted equational theory $(\Omega \uplus \Delta, E_\Delta)$ according to the above-described theory transformation are the following: (i) the equations E_Δ are size-reducing and therefore terminating and, having no critical pairs, they are also confluent; (ii) the decomposition $(\Omega \uplus \Delta, \emptyset, R(E_\Delta))$ has the finite variant property, as can be easily checked by remarking that for each y of sort B_c $sel_i^c(y)$ has just two variants, namely, $(sel_i^c(y), id)$, and $(x_i, \{y \mapsto c(x_1, \dots, x_n)\})$; (iii) $(\Omega, \emptyset, \emptyset)$ is the constructor decomposition of $(\Omega \uplus \Delta, \emptyset, R(E_\Delta))$ and therefore is OS-compact. In particular, by setting $\Omega_\Pi = \emptyset$ and therefore skipping Step 2, the decision procedure we have proved correct specializes to a decision procedure for satisfiability of QF $\Omega \uplus \Delta$-formulas in $T_{\Omega \uplus \Delta / E_\Delta}$ for any recursive data structure with selectors.[4]

We can however increase the expressive power of the language of recursive data structures by adding *sort predicates*. Specifically, we add a fresh new sort *Pred* with constant *tt*, and for each subsort $B_c < B$ a predicate $\_:B_c : B \rightarrow Pred$ defined by the equation $y:B_c = tt$, with y of sort B_c. Since it is easy to check that such predicate-defining equations are terminating and confluent, and each term $z:B_c$ with z of sort B has exactly two variants, we obtain in this way an FVP decomposition of the form $(\Omega \uplus \Omega_\Pi \uplus \Delta, \emptyset, R(E_\Delta) \uplus R(E_\Pi))$ with E_Π the above sort-predicate-defining equations. Furthermore, this decomposition satisfies conditions (1)–(3), so that the inductive decidability decision procedure for

[4] We assume that all sorts are non-empty. For example, the sort *Elt* for lists is assumed already instantiated to some data structure of elements.

initial algebras with user-definable predicates can be applied. The only condition that needs some explanation is condition (3). Indeed, let a sort B have different constructors $c, c_1, \ldots, c_m$ where some of the $c_1, \ldots, c_m$ could be constants. Then the negative patterns for the sort predicate $\_:B_c$ are precisely the patterns $c_j(z_1, \ldots, z_{n_j}):B_c \neq tt$, $1 \leqslant j \leqslant m$. In summary, we have:

Theorem 7. *Let Ω_0 be a many-sorted signature of constructors defining an initial algebra T_{Ω_0} axiomatizing any kind of recursive data structures. Let Ω be its corresponding order-sorted version and $(\Omega \uplus \Omega_\Pi \uplus \Delta, \varnothing, R(E_\Delta) \uplus R(E_\Pi))$ its extension by both selectors Δ and sort predicates Ω_Π according to the above transformation. Then QF satisfiability in the initial algebra $T_{\Omega \uplus \Omega_\Pi \uplus \Delta / E_\Delta \uplus E_\Pi}$ is 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: $0 : NzNat \neq tt$ and $nil : NeList \neq tt$.

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 as a part of a new prototype tool. The implementation consists of 11 new Maude modules (from 17 total), 2345 new lines of code, and uses the Maude's `META-LEVEL` to carry out the steps of the procedure. We have also developed a Maude interface to ease the definition of properties and patterns as equations.

Example 8. We show below the running example of the paper in Maude. The union of modules `ACU-NAT-FUN` and `NAT-SET` specifies the equational theory *Nat-Set* presented in Example 1. `NAT-SET` itself specifies the OS-compact constructor decomposition $\mathcal{R}_{\Omega_c}$ in Example 2.

```
fmod ACU-NAT is
  sort Natural .

  op 0   : -> Natural [ctor] .
```

```
  op 1   : -> Natural [ctor] .
  op _+_ : Natural Natural -> Natural [ctor assoc comm id: 0] .
endfm

fmod ACU-NAT-FUN is
  pr ACU-NAT .

  op max : Natural Natural -> Natural [comm] .
  op min : Natural Natural -> Natural [comm] .
  op _-_ : Natural Natural -> Natural .  *** monus

  vars N M : Natural .

  eq max(N,N + M) = N + M [variant] .
  eq min(N,N + M) = N [variant] .

  eq N - (N + M) = 0 [variant].
  eq (N + M) - N = M [variant] .
endfm

fmod NAT-SET is
  pr ACU-NAT .

  sort NaturalSet .
  sort Pred .

  subsort Natural < NaturalSet .

  op mt  : -> NaturalSet [ctor] .
  op _,_ : NaturalSet NaturalSet -> NaturalSet [ctor assoc comm] .
  op tt  : -> Pred [ctor] .

  *** set containment
  op _=C_ : NaturalSet NaturalSet -> Pred [ctor] .

  vars NS NS' : NaturalSet .

  *** identity of set union
  eq NS , mt = NS [variant] .
  *** idempotency of set union
  eq NS , NS = NS [variant] .
  *** idempotency of set union
  eq NS , NS , NS' = NS , NS' [variant] .

  eq mt =C NS = tt [variant] .
```

```
  eq NS =C NS = tt [variant] .
  eq NS =C NS , NS' = tt [variant] .
endfm
```

The extension *NatSetPreds* of the theory *NatSet* in Example 4 is specified by the NAT-SET-PREDS module.

```
fmod NAT-SET-PREDS is
  pr NAT-SET .

  *** strict order
  op _>_ : Natural Natural -> Pred [ctor] .
  *** sort predicates
  op natural : NaturalSet -> Pred [ctor] .
  op even : NaturalSet -> Pred [ctor] .
  op odd : NaturalSet -> Pred [ctor] .

  vars N M : Natural .

  eq N + M + 1 > N = tt [variant] .

  eq natural(N) = tt [variant] .

  eq even(N + N) = tt [variant] .

  eq odd(N + N + 1) = tt [variant] .
endfm
```

A QF formula we want to test for variant satisfiability is specified as a non-executable equations labeled with the `conjecture` keyword. The negative patterns of each user-defined predicate outside the OS-compact subtheory are also specified as non-executable equations labeled with the `nPattern` keyword.

Suppose we want to prove the inductive validity of the following formulae:

1. $\texttt{N - M} = \texttt{0} \Leftrightarrow (\texttt{M > N} = \texttt{tt} \vee \texttt{N} = \texttt{M})$, and
2. $\texttt{natural(NS)} = \texttt{tt} \Rightarrow (\texttt{even(NS)} = \texttt{tt} \vee \texttt{odd(NS)} = \texttt{tt})$.

This is equivalent to proving that each conjunction in the DNF of each negated formula is unsatisfiable. Therefore, our input conjectures are:

1. $(\texttt{N - M} = \texttt{0} \wedge \texttt{M > N} \neq \texttt{tt} \wedge \texttt{N} \neq \texttt{M}) \vee (\texttt{N - M} \neq \texttt{0} \wedge \texttt{M > N} = \texttt{tt}) \vee (\texttt{N - M} \neq \texttt{0} \wedge \texttt{N} = \texttt{M})$ for the first formula, and
2. $\texttt{natural(NS)} = \texttt{tt} \wedge \texttt{even(NS)} \neq \texttt{tt} \wedge \texttt{odd(NS)} \neq \texttt{tt}$ for the second formula.

The three conjunctions of the first formula correspond to the equations `prop1a`, `prop1b` and `prop1c` in module `NAT-SET-PRED-CONJECTURES-PATTERNS`, and the second formula corresponds to the equation `prop2`.

We also specify in module `NAT-SET-PRED-CONJECTURES-PATTERNS` the negative patterns described in Example 5.

```
mod NAT-SET-PREDS-CONJECTURES-PATTERNS is
  pr ATOM-MAGMA-SET .
  pr PATTERN .
  pr NAT-SET-PREDS .
  pr NAT-FUN .

*** patterns
  op neg-gr      : Natural Natural -> Pattern .
  op neg-even    : NaturalSet -> Pattern .
  op neg-odd     : NaturalSet -> Pattern .
  op neg-natural : NaturalSet -> Pattern .

*** formulae
  op prop1a : Natural Natural -> AtomMagma .
  op prop1b : Natural Natural -> AtomMagma .
  op prop1c : Natural Natural -> AtomMagma .
  op prop2  : NaturalSet -> AtomMagma .

  vars N M K : Natural .
  var NS : NaturalSet .

  eq prop1a(N,M) = (N - M = 0) , (M > N /= tt)
       , (N /= M) [nonexec label conjecture] .
  eq prop1b(N,M) = (N - M /= 0)
       , (M > N = tt) [nonexec label conjecture] .
  eq prop1c(N,M) = (N - M /= 0)
       , (N = M) [nonexec label conjecture] .

  eq prop2(NS) = (natural(NS) = tt) , (even(NS) /= tt)
       , (odd(NS) /= tt) [nonexec label conjecture] .

  eq neg-gr(N,M) = N > N + M
       | (empty).AtomMagma [nonexec label nPattern] .
  eq neg-even(N) = even(N + N + 1)
       | (empty).AtomMagma [nonexec label nPattern] .
  eq neg-even((N , NS)) = even((N, NS))
       | ((N =C NS /= tt) , (NS /= mt)) [nonexec label nPattern] .
  eq neg-odd(N) = odd(N + N)
       | (empty).AtomMagma [nonexec label nPattern] .
  eq neg-odd((N , NS)) = odd((N, NS))
       | ((N =C NS /= tt) , (NS /= mt)) [nonexec label nPattern] .
  eq neg-natural((N , NS)) = natural((N, NS))
       | ((N =C NS /= tt) , (NS /= mt)) [nonexec label nPattern] .
endm
```

In the execution, our main function uses four arguments:

1. the whole theory $(\Sigma, E \cup U)$,
2. the subtheory that contains the predicates,
3. the constructor subtheory $(\Omega, E_\Omega \cup B_\Omega)$ which is OS-compact, and
4. the patterns and the conjectures.

```
fmod NAT-SET-PREDS-THEORY is
  pr NAT-SET-PREDS .
  pr NAT-FUN .
endfm

fmod NAT-SET-PREDS-INTERFACE is
  pr NAT-SET-PREDS-THEORY .
  pr NU-ITP-INTERFACE .
endfm

red in NAT-SET-PREDS-INTERFACE :
  initTheory(upModule('NAT-SET-PREDS-THEORY,true)
    , upModule('NAT-SET-PREDS,true)
    , upModule('NAT-SET,true)
    , upModule('NAT-SET-PREDS-CONJECTURES-PATTERNS
        ,false)) .
```

The three steps of the variant satisfiability procedure take advantage of Maude's `META-LEVEL` functions:

- in the **unification** step we use the `metaVariantUnify` function to obtain the different variants and `metaReduce` to obtain the normal form of the conjunction, and
- in the **predicate elimination** step we use the `metaGetIrredundantVariant` function to then filter out the constructor Ω_c-variants, and the `metaDisjointUnify` function to obtain the different instances of the negative patterns.
- in the **computation of $\Omega_c^\wedge$-variants** step we use `metaGetIrredundantVariant` function to then filter out constructor $\Omega_c^\wedge$-variants; the current prototype assumes that all sorts are infinite.

Example 9. Continuing with the above example, consider the check that `prop1a` is unsatisfiable:

1. The input conjecture is:

$$(\texttt{N - M} = \texttt{0} \wedge \texttt{M > N} \neq \texttt{tt} \wedge \texttt{N} \neq \texttt{M})$$

2. After the **unification** step, we obtain

$$\texttt{(V2 + V3) > V2} \neq \texttt{tt} \wedge \texttt{V2} \neq \texttt{V2 + V3}$$

   where `V2` and `V3` are variables of sort `Natural`.

3. Applying the Π-**elimination** step, we obtain:

$$\texttt{V4} \neq \texttt{V4 + 0}$$

where `V4` is a variable of sort `Natural`. After normalization, the formula becomes B_{Ω_c}-inconsistent and therefore unsatisfiable.

Similarly, we can check the unsatisfiability of all other conjunctions in this example, thus proving the validity of the two inductive theorems.

6 Related Work and Conclusions

The original paper proposing the concepts of variant and FVP is [9]. FVP ideas have been further advanced in [14,6,4,5]. Variant satisfiability has been studied on [24,25,31]. In relation to that work, the main contribution of this paper is the extension of variant satisfiability to handle user-definable predicates.

Other theory-generic satisfiability approaches include: (i) the superposition-based one, e.g., [22,2,21,23,3,1,12,32], 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 [11], 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 can be seen a 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 [24,25] and implemented in [31] 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 [24,31].

References

1. Armando, A., Bonacina, M.P., Ranise, S., Schulz, S.: New results on rewrite-based satisfiability procedures. ACM Trans. Comput. Log. 10(1) (2009)
2. Armando, A., Ranise, S., Rusinowitch, M.: A rewriting approach to satisfiability procedures. Inf. Comput. 183(2), 140–164 (2003)
3. Bonacina, M.P., Echenim, M.: On variable-inactivity and polynomial $\mathcal{T}$-satisfiability procedures. J. Log. Comput. 18(1), 77–96 (2008)

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. 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`
6. Ciobaca., S.: Verification of Composition of Security Protocols with Applications to Electronic Voting. Ph.D. thesis, ENS Cachan (2011)
7. 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
8. Comon, H.: Complete axiomatizations of some quotient term algebras. Theor. Comput. Sci. 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. Dross, C., Conchon, S., Kanig, J., Paskevich, A.: Adding decision procedures to SMT solvers using axioms with triggers. J. Autom. Reasoning 56(4), 387–457 (2016)
12. Echenim, M., Peltier, N.: An instantiation scheme for satisfiability modulo theories. J. Autom. Reasoning 48(3), 293–362 (2012)
13. Ehrig, H., Mahr, B.: Fundamentals of Algebraic Specification 1. Springer (1985)
14. Escobar, S., Sasse, R., Meseguer, J.: Folding variant narrowing and optimal variant termination. J. Algebraic and Logic Programming 81, 898–928 (2012)
15. 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)
16. 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)
17. 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)
18. Hendrix, J., Ohsaki, H., Viswanathan, M.: Propositional tree automata. In: RTA 2006. Lecture Notes in Computer Science, vol. 4098, pp. 50–65. Springer (2006)
19. 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)
20. 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)
21. 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)
22. Lynch, C., Morawska, B.: Automatic decidability. In: Proc. LICS 2002. p. 7. IEEE Computer Society (2002)
23. Lynch, C., Tran, D.: Automatic decidability and combinability revisited. In: Proc. CADE 2007. vol. 4603, pp. 328–344. Springer LNCS (2007)
24. Meseguer, J.: Variant-based satisfiability in initial algebras. Tech. Rep. `http://hdl.handle.net/2142/88408`, University of Illinois at Urbana-Champaign (November 2015)

25. Meseguer, J.: Variant-based satisfiability in initial algebras. In: Artho, C., Ölveczky, P. (eds.) Proc. FTSCS 2015. pp. 1–32. Springer CCIS 596 (2016)
26. Meseguer, J.: Membership algebra as a logical framework for equational specification. In: Proc. WADT'97. pp. 18–61. Springer LNCS 1376 (1998)
27. Meseguer, J.: Strict coherence of conditional rewriting modulo axioms. Theor. Comput. Sci. 672, 1–35 (2017)
28. Meseguer, J., Goguen, J.: Order-sorted algebra solves the constructor-selector, multiple representation and coercion problems. Information and Computation 103(1), 114–158 (1993)
29. Nelson, G., Oppen, D.C.: Simplification by cooperating decision procedures. ACM Trans. Program. Lang. Syst. 1(2), 245–257 (1979)
30. Shostak, R.E.: Deciding combinations of theories. Journal of the ACM 31(1), 1–12 (Jan 1984)
31. 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)
32. Tushkanova, E., Giorgetti, A., Ringeissen, C., Kouchnarenko, O.: A rule-based system for automatic decidability and combinability. Sci. Comput. Program. 99, 3–23 (2015)

A Equivalences for Negative Patterns in *NatSetPreds*

In case the positive and negative patterns for user-definable predicates are linear, the negative patterns are unconditional, and the constructors are free modulo axioms like associativity and/or commutativity and/or identity, tree automata techniques such as those in [18] can be used to automatically prove the two desired semantic equivalences. Since the *NatSetPreds* example is outside this case, a different proof is needed. However, the proof that, for $\mathcal{R} = \mathit{NatSetPreds}$, $\mathcal{C}_{\mathcal{R}}$ satisfies the first equivalence:

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

can be automated using folding variant narrowing. Specifically, it is enough to show that any non-identity variants of a negative constrained pattern —i.e., those corresponding to a reducible instance of the pattern— have substitutions that violate the pattern's constraint. Since this is a general technique that can always be used to automate the proof of the first equivalence, we include all the details to illustrate the general method.

For the pattern $N > N + M \neq tt$, Maude2.7.1 returns only the identity variant:

```
Maude> get irredundant variants in NAT-SET-PREDS : N > N + M .

Variant #1
Pred: #1:Nat > #1:Nat + #2:Nat
N --> #1:Nat
M --> #2:Nat

No more variants.
```

For the pattern $even(N + N + 1) \neq tt$, we again only get the identity variant:

```
Maude> get irredundant variants in NAT-SET-PREDS : even(N + N + 1) .

Variant #1
Pred: even(1 + #1:Nat + #1:Nat)
N --> #1:Nat

No more variants.
```

For the pattern $(N \subseteq NS \neq tt \ \wedge \ NS \neq \emptyset) \Rightarrow even(N, NS) \neq tt$, we get:

```
Maude> get irredundant variants in NAT-SET-PREDS : even(N,NS) .

Variant #1
Pred: even(#1:Nat,#2:NatSet)
N --> #1:Nat
NS --> #2:NatSet

Variant #2
Pred: even(%1:Nat)
N --> %1:Nat
NS --> mt

Variant #3
Pred: even(%1:Nat)
N --> %1:Nat
NS --> %1:Nat

Variant #4
Pred: even(%1:Nat,%2:NatSet)
N --> %1:Nat
NS --> %1:Nat,%2:NatSet

Variant #5
Pred: tt
N --> #1:Nat + #1:Nat
NS --> mt

Variant #6
Pred: tt
N --> #1:Nat + #1:Nat
NS --> #1:Nat + #1:Nat

No more variants.
```

where the constraint $NS \neq \emptyset$ is violated by variants 2 and 5, and the constraint $N \subseteq NS \neq tt$ is violated by variants 3, 4 and 6.

For the pattern $odd(N + N) \neq tt$, we again only get the identity variant:

```
get irredundant variants in NAT-SET-PREDS : odd(N + N) .

Variant #1
Pred: odd(#1:Nat + #1:Nat)
N --> #1:Nat

No more variants.
```

For the pattern $(N \subseteq NS \neq tt \wedge NS \neq \emptyset) \Rightarrow odd(N, NS) \neq tt$, again all non-identity variants violate the pattern's constraint.

```
Maude> get irredundant variants in NAT-SET-PREDS : odd(N,NS) .

Variant #1
Pred: odd(#1:Nat,#2:NatSet)
N --> #1:Nat
NS --> #2:NatSet

Variant #2
Pred: odd(%1:Nat)
N --> %1:Nat
NS --> mt

Variant #3
Pred: odd(%1:Nat)
N --> %1:Nat
NS --> %1:Nat

Variant #4
Pred: odd(%1:Nat,%2:NatSet)
N --> %1:Nat
NS --> %1:Nat,%2:NatSet

Variant #5
Pred: tt
N --> 1 + #1:Nat + #1:Nat
NS --> mt

Variant #6
Pred: tt
N --> 1 + #1:Nat + #1:Nat
NS --> 1 + #1:Nat + #1:Nat
```

```
No more variants.
```

Finally, for the pattern $(N \subseteq NS \Doteq tt \wedge NS \Doteq \varnothing) \Rightarrow (N, NS) : Nat \Doteq tt$, all non-identity variants again violate the pattern's constraint.

```
Maude> get irredundant variants in NAT-SET-PREDS : (N,NS) :Nat .

Variant #1
Pred: (#1:Nat,#2:NatSet) :Nat
N --> #1:Nat
NS --> #2:NatSet

Variant #2
Pred: tt
N --> %1:Nat
NS --> mt

Variant #3
Pred: tt
N --> %1:Nat
NS --> %1:Nat

Variant #4
Pred: (%1:Nat,%2:NatSet) :Nat
N --> %1:Nat
NS --> %1:Nat,%2:NatSet

No more variants.
```

Since we have already proved that for each predicate the constrained negative patterns and the patterns corresponding to the lefthand sides of the equations defining each predicate have mutually disjoint instances (the first irreducible, and the second reducible), we can prove that the second equivalence

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

holds in $\mathcal{C}_{\mathcal{R}}$ by showing that for each predicate p the positive and negative patterns together are "sufficiently complete," in the sense that they cover all instances of p by irreducible ground arguments that are either reducible by the equations for p or are irreducible.

Consider first the case of 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 $J > K$, and (iv) $J \subseteq K$ holds iff $(\exists M)\ K = M + K$.

Consider next the case of the $\_ : Nat$ predicate, which has the positive pattern $N : Nat$ with N of sort Nat and the negative pattern $(N \subseteq NS \neq tt \wedge NS \neq \emptyset) \Rightarrow (N, NS) : Nat \neq tt$. Note that any irreducible, non-singleton finite set of natural numbers 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 $1 \leqslant i < j \leqslant k$. And 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 pattern's constraint.

Since for the *even* and *odd* predicates the conditional negative patterns are completely analogous to the one for the $\_ : Nat$ predicate and exactly cover the case of irreducible non-singleton sets of natural numbers, we only need to show that, in each case, the positive pattern and the unconditional negative one cover all natural numbers. Since both cases are entirely similar, it is enough to show this for the case of the *even* predicate, where the positive pattern is $even(N + N)$ and the negative one is $even(M + M + 1)$. 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.