Order-Sorted Rewriting and Congruence Closure

José Meseguer(✉)

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

Abstract. Order-sorted type systems supporting inheritance hierarchies and subtype polymorphism are used in theorem proving, AI, and declarative programming. The satisfiability problems for the theories of: (i) order-sorted uninterpreted function symbols, and (ii) of such symbols *modulo* a subset Δ of associative-commutative ones are *reduced* to the *unsorted* versions of such problems at no extra computational cost. New results on order-sorted rewriting are needed to achieve this reduction.

Keywords: Order-sorted rewriting · Congruence closure · Satisfiability

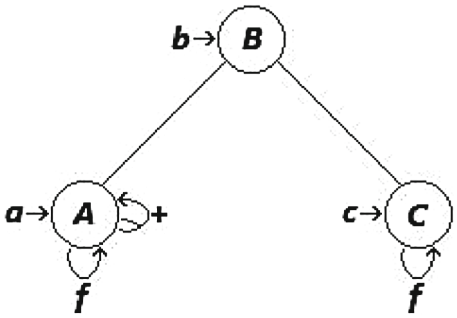
1 Introduction

For greater expressiveness and efficiency, type systems supporting inheritance hierarchies and subtype polymorphism are used in many areas such as resolution theorem proving, e.g., [26,32], declarative logic and rule-based languages, e.g., [4,9,10,29], and artificial intelligence, e.g., [8,29]. Order-sorted (OS) equational logic, e.g., [15,21], is a logical framework supporting inheritance hierarchies and subtype polymorphism widely used for these purposes. Therefore, the development of *decision procedures* for OS theories is of interest in all these areas. I focus here on decision procedures for the OS theory of *uninterpreted function symbols*, which in an unsorted setting is decided by congruence closure algorithms [7,24,27]. However, for greater expressiveness one can allow some of the function symbols, say in a subsignature $\Delta \subseteq \Sigma$, to be *interpreted* by some axioms B_Δ. For example, for an unsorted subsignature $\Delta \subseteq \Sigma$ of binary function symbols, congruence closure algorithms *modulo* the axioms AC_Δ, asserting the associativity and commutativity of all symbols in Δ have been given in [2,19,22]. Therefore, I also study satisfiability in the OS theory (Σ, AC_Δ) of *uninterpreted function symbols* Σ *modulo* AC_Δ.

The most obvious approach would be to develop an *order-sorted* congruence closure algorithm along the lines of [11] and then extended it to the modulo AC case. However, the main, somewhat surprising message of this paper is that such OS congruence closure algorithms *are not needed at all*: the already existing and efficient *unsorted* congruence closure algorithms in [7,24,27] and congruence closure modulo AC_Δ in [2,19,22] and tools supporting them can be reused *without change* and *at no extra cost* to solve the corresponding OS satisfiability problems.

B. Jacobs and C. Löding (Eds.): FOSSACS 2016, LNCS 9634, pp. 493–509, 2016.
DOI: 10.1007/978-3-662-49630-5_29

A Simple Example. Consider the following order-sorted signature Σ



with sorts A, B, C, subsorts $A, C < B$, f subsort-polymorphic with typings $f : A \to A$ and $f : C \to C$, and a binary $+$ with typing $+ : A\,A \to A$. Its so-called *theory of uninterpreted function symbols* is just the order-sorted equational theory $(\Sigma, \emptyset)$ with empty set of equations, whose class of models, $\mathbf{OSAlg}_\Sigma$, is that of *all* order-sorted Σ-algebras detailed in Sect. 2. Is the formula

$$(\flat) \qquad a = b \wedge b = c \wedge f(f(a)) = f(a) \wedge a + f(f(a)) \neq f(a) + a$$

$(\Sigma, \emptyset)$-*satisfiable*? The standard way to answer this question if Σ were unsorted would be to: (1) compute the congruence closure of the first three equations; and (2) test the last inequality using such a congruence closure. Since, as pointed out in [2,12,16], unsorted congruence closure algorithms are *ground Knuth-Bendix* completion algorithms [18], an obvious way to try to answer this question would be to try to *complete* the first three equations into an equivalent set of confluent and terminating rewrite rules. But this runs into serious trouble. An order-sorted Knuth-Bendix completion algorithm such as [13] will orient $a = b$ and $b = c$ as $b \to a$ and $b \to c$ because rules must be *sort-decreasing*, i.e., rewrite to a term of equal or lower sort. This then generates the critical pair $a = c$, which is *unorientable*, so completion fails. Notice also that *replacement of equals by equals* does not hold in an order-sorted setting: from $a = b$ we *cannot* derive $f(a) = f(b)$, because $f(b)$ doesn't type. These difficulties were clearly felt by the authors of [11], the only order-sorted congruence closure algorithm I am aware of, which is quite complex and is *not* a Knuth-Bendix completion. They say:

> *An approach using rewriting* [...] *fails due to the well-known problem that rewriting with order-sorted rewrite rules may create ill-typed terms.*

Let us now widen the problem into one of *satisfiability modulo AC* by making the $+$ symbol associative-commutative. That is, we consider the axioms $AC_+ = \{x+y = y+x, (x+y)+z = x+(y+z)\}$, with x, y, z of sort A, and ask: is the formula $(\flat)$ (Σ, AC_+)-*satisfiable*? For this case, I am not aware of any order-sorted AC-congruence closure algorithm, but unsorted, ground-AC-completion-based ones exist [2,19,22]. The trouble, again, is that *order-sorted* AC-completion as in [13] fails miserably in the *same* way ($a = c$ cannot be oriented).

Wouldn't it be nice if we could *completely ignore* all sort information in the above two OS satisfiability problems and solve them as *unsorted* problems using standard (and efficient!) congruence closure [7,24,27] and congruence closure modulo AC [2,19,22] algorithms? If this reduction method were *sound*, we could easily settle the $(\Sigma, \emptyset)$- and (Σ, AC_+)-satisfiability of $(\flat)$: the confluent and terminating rules $R = \{a \to b, c \to b, f(f(b)) \to f(b)\}$ play the role of a "congruence closure" for the first three equations, and also of an AC_+-congruence closure. Since the disequality $a + f(f(a)) \neq f(a) + a$ reduces to $b + f(b) \neq f(b) + b$, the formula $(\flat)$ is $(\Sigma, \emptyset)$-satisfiable. However, since $b + f(b) =_{AC_+} f(b) + b$, $(\flat)$ is (Σ, AC_+)-*unsatisfiable*. But is this *reduction* to unsorted satisfiability *sound*?

Initial Algebra Semantics of Uninterpreted Satisfiability. Ignoring the sort information of an OS signature Σ is captured by a signature map $u : \Sigma \ni (f : s_1 \dots s_n \to s) \mapsto (f : U \stackrel{n}{\ldots} U \to U) \in \Sigma^u$, where U is the single "universe" sort in the unsorted signature Σ^u. As further detailed at the end of Sect. 2, u induces a *reduct* map of algebras in the opposite direction, $\_|_u : \mathbf{Alg}_{\Sigma^u} \ni A \mapsto A|_u \in \mathbf{OSAlg}_\Sigma$, making each unsorted algebra A into and order-sorted one $A|_u$, and such that for a set of ground OS Σ-equations E we have the equivalence: $A|_u \models E \Leftrightarrow A \models E$. In particular, the E-initial unsorted Σ^u-algebra $T_{\Sigma^u/E}$ is mapped to the OS Σ-algebra $T_{\Sigma^u/E}|_u$ and, since $T_{\Sigma^u/E}|_u \models E$, there is a unique OS homomorphism $h : T_{\Sigma/E} \to T_{\Sigma^u/E}|_u$ from the E-initial OS Σ-algebra $T_{\Sigma/E}$.

But the proof of Theorem 5 shows that, for equations E and disequations D, the conjunction $\bigwedge E \wedge \bigwedge D$ is satisfiable iff $T_{\Sigma(C)/E} \models \bigwedge E \wedge \bigwedge D$, where the variables C of $E \cup D$ are seen as *fresh new constants* added to Σ to get a supersignature $\Sigma(C) \supseteq \Sigma$, so that $\bigwedge E \wedge \bigwedge D$ becomes a *ground* formula. This gives us, in model-theoretic terms, the key to verify the soundness of the hoped-for *reduction* of the satisfiability for the theory of OS uninterpreted function symbols to that of the unsorted theory of uninterpreted function symbols: this reduction method will be *sound* if and only if the OS homomorphism $h : T_{\Sigma(C)/E} \to T_{\Sigma(C)^u/E}|_u$ is *injective*. In proof-theoretic terms this injectivity will hold if and only if for all ground Σ-equation $u = v$ we have the equivalence: $(\Sigma, E) \vdash u = v \Leftrightarrow (\Sigma^u, E) \vdash u = v$. The $(\Rightarrow)$ direction is obvious, but the $(\Leftarrow)$ direction is a non-trivial new result that follows from several *conservativity theorems* that I prove in Sects. 3.2 and 4.1 by factoring the signature map $u : \Sigma \to \Sigma^u$ through a sequence $\Sigma \hookrightarrow \Sigma^\square \to \widehat{\Sigma} \to \Sigma^u$ of increasingly simpler order-sorted, many-sorted and finally unsorted signatures and relating equational and rewriting deductions at all these levels.

The Plot Thickens. The soundness of the hoped-for reduction to the unsorted case for satisfiability modulo AC_Δ is a thornier issue. As before, the reduction will be sound if and only if for ground Σ-equations E the unique Σ-homomorphism $h : T_{\Sigma/E \cup AC_\Delta} \to T_{\Sigma^u/E \cup AC_{\Delta^u}}|_u$ from the initial $E \cup AC_\Delta$-algebra $T_{\Sigma/E \cup AC_\Delta}$ is *injective*. But some of the conservativity theorems along the above sequence of signature maps $\Sigma \hookrightarrow \Sigma^\square \to \widehat{\Sigma} \to \Sigma^u$ needed to make h injective actually *break down* in the AC_Δ case. The problem has to do with the translation of the equations AC_Δ along these signature maps. At the unsorted

level of Σ^u the translated equations AC_{Δ^u}, are *more general* and therefore *identify more terms* than the original OS equations AC_Δ. Consider a simple example: the equation $a + b = b + a$ does not type in our example signature Σ, but it types in the supersignature $\Sigma^\square \supseteq \Sigma$, which for our running example is depicted in Sect. 3.1. The AC equations AC_Δ in our example are just associativity and commutativity of $+ : A\ A \to A$ and therefore *apply only* to terms of sort A. Instead, the AC equations AC_{Δ^u} are unsorted, and *apply to all terms*. This means that $a + b =_{AC_{\Delta^u}} b + a$, but since b does not have sort A, we have $a + b \neq_{AC_\Delta} b + a$. It also means that the homomorphism $h' : T_{\Sigma^\square/E \cup AC_\Delta} \to T_{\Sigma^u/E \cup AC_{\Delta^u}}|_u$ in general is *not* injective. However, all hope is not lost. As a direct consequence of Corollary 2 in Sect. 3.2, there is an isomorphism $\alpha : T_{\Sigma/E \cup AC_\Delta} \cong T_{\Sigma^\square/E \cup AC_\Delta}|_\Sigma$ to the Σ-*reduct* of $T_{\Sigma^\square/E \cup AC_\Delta}$ and this shows that the homomorphism $h : T_{\Sigma/E \cup AC_\Delta} \to T_{\Sigma^u/E \cup AC_{\Delta^u}}|_u$ that we need to prove injective for the reduction to be sound is up to isomorphism a *restriction* of h' to $T_{\Sigma/E \cup AC_\Delta}$, which *could* be injective even if h' is not. Lemma 3 in Sect. 4.1 and the highly non-trivial Theorem 8 in Sect. 5 save the day: it follows from them that h is indeed injective and the reduction is also sound for the AC case. To the best of my knowledge the results on reducing order-sorted to unsorted satisfiability and on order-sorted rewriting and equality are new.

The paper is organized as follows. After some preliminaries in Sect. 2, the new results on order-rewriting and equality are given in Sect. 3. The reductions of satisfiability in the theory of OS uninterpreted function symbols (resp. OS uninterpreted function symbols modulo AC) to satisfiability in their respective unsorted theories is given in Sect. 4 (resp. Sect. 5). Related work and conclusions are discussed in Sect. 6. Due to space limitations no proofs are given; they can be found in the Technical Report [20].

2 Preliminaries on Order-Sorted Algebra

The following material is adapted from [21], which generalizes [15]. It summarizes the basic notions of order-sorted algebra needed in the rest of the paper.

Definition 1. *A* many-sorted signature *is a pair $\Sigma = (S, \Sigma)$, with S a set of sorts, and Σ and $S^* \times S$-indexed set $\Sigma = \{\Sigma_{w,s}\}_{w,s \in S^* \times S}$ of operation symbols, where S^* denotes the free monoid generated by S. We denote each $f \in \Sigma_{w,s}$ as $f : w \to s$. In particular, a* constant *of sort s is an operation $a : \epsilon \to s$, with ϵ the empty word.*

An order-sorted (OS) signature *is a triple $\Sigma = (S, \leq, \Sigma)$ with $(S, \leq)$ a poset and (S, Σ) a many-sorted signature. $\widehat{S} = S/\equiv_\leq$, the quotient of S under the equivalence relation $\equiv_\leq = (\leq \cup \geq)^+$, is called the set of* connected components *of $(S, \leq)$. Note that a* many-sorted signature *Σ is the special case where the poset $(S, \leq)$ is discrete, i.e., $s \leq s'$ iff $s = s'$.*

The order $\leq$ and equivalence $\equiv_\leq$ are extended to sequences of same length in the usual way, e.g., $s'_1 \ldots s'_n \leq s_1 \ldots s_n$ iff $s'_i \leq s_i$, $1 \leq i \leq n$. Σ is called

sensible[1] *if for any two* $f : w \to s, f : w' \to s' \in \Sigma$, *with* w *and* w' *of same length, we have* $w \equiv_{\leq} w' \Rightarrow s \equiv_{\leq} s'$.

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

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

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

Definition 2. *For* $\Sigma = (S, \Sigma)$ *a many-sorted signature, a* Σ-algebra *is an* S*-indexed set* $A = \{A_s\}_{s \in S}$ *together with an assignment of: (i) to each constant* $a : \epsilon \to s$ *of sort* s *an element* $A_a \in A_s$, *and (ii) to each operation* $f : w \to s$, *with* $w = s_1 \ldots s_n$, $n \geq 1$, *a function* $A_{f:w \to s} : A^w \to A_s$, *where, by convention,* $A^{s_1 \ldots s_n} = A_{s_1} \times \ldots \times A_{s_n}$.

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

- *whenever* $s \leq s'$, *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* $A_{f:w \to s}(\overline{a}) = A_{f:w' \to s'}(\overline{a})$.

A many-sorted Σ-homomorphism $h : A \to B$ *is an* S*-indexed family of functions* $h = \{h_s : A_s \to B_s\}_{s \in S}$ *such that: (i) for* $a : \epsilon \to s$, $h_s(A_a) = B_a$, *and (ii) for* $f : w \to s$ *with* $w \neq \epsilon$, $A_f; h_s = h^w; B_f$.

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* $h_s(a) = h_{s'}(a)$. *We call* h injective, *resp.* surjective, *resp.* bijective, *iff for each* $s \in S$ h_s *is injective, resp. surjective, resp. bijective. We 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 [21]. *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* $a \in T_{\Sigma,s}$ *(*ϵ *denotes the empty string),*
- *if* $t \in T_{\Sigma,s}$ *and* $s \leq s'$ *then* $t \in T_{\Sigma,s'}$,
- *if* $f : s_1 \ldots s_n \to s$ *and* $t_i \in T_{\Sigma,s_i}$ $1 \leq i \leq n$, *then* $f(t_1, \ldots, t_n) \in T_{\Sigma,s}$,

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

For $[s] \in \widehat{S}$, $T_{\Sigma,[s]}$ denotes the set $T_{\Sigma,[s]} = \bigcup_{s' \in [s]} T_{\Sigma,s'}$. Similarly, T_{Σ} will (ambiguously) denote both the above-defined S-sorted set and the set $T_{\Sigma} =$

[1] The notion of a sensible signature is a *minimal syntactic requirement* to avoid excessive ambiguity. For example, a many-sorted signature Σ with sorts A, B and C, constant $a : \epsilon \to A$ and operations $f : A \to B$ and $f : A \to C$ is not sensible and therefore is intrinsically ambiguous: the term $f(a)$ has both sorts B and C, which are completely different sorts.

$\bigcup_{s \in S} T_{\Sigma,s}$. We say that an OS signature Σ *has 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 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-homomorphim, $\_\alpha : T_\Sigma(X) \longrightarrow A$ extending α, i.e., such that for each $s \in S$ and $x \in X_s$ we have $x\alpha_s = \alpha_s(x)$.*

The first-order language of *equational Σ-formulas*[2] 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, we define the *satisfaction relation* $A, \alpha \models \varphi$ inductively as usual: for atoms, $A, \alpha \models t = t'$ iff $t\alpha = t'\alpha$; for Boolean connectives it is the corresponding Boolean combination of the satisfaction relations for subformulas; and for quantifiers: $A, \alpha \models (\forall x : s)\ \varphi$ (resp. $A, \alpha \models (\exists x : s)\ \varphi$) holds iff for all $a \in A_s$ (resp. some $a \in A_s$) we have $A, \alpha \uplus \{(x{:}s, a)\} \models \varphi$, where the assignment $\alpha \uplus \{(x{:}s, a)\}$ extends α by mapping $x{:}s$ to a. Finally, $A \models \varphi$ holds iff $A, \alpha \models \varphi$ holds for each $\alpha \in [Y \to A]$, where $Y = fvars(\varphi)$. We say that φ is *valid* (or *true*) in A iff $A \models \varphi$. We say that φ is *satisfiable* in A iff $\exists \alpha \in [Y \to A]$ such that $A, \alpha \models \varphi$, where $Y = fvars(\varphi)$.

An *order-sorted 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* $T_{\Sigma/E}$ [21], further discussed in Sect. 3. Given $T = (\Sigma, E)$ and $\varphi \in Form(\Sigma)$, we call φ *T-valid*, written $E \models \varphi$, iff $A \models \varphi$ for each $A \in \mathbf{OSAlg}_{(\Sigma,E)}$. We call φ *T-satisfiable* iff there exists $A \in \mathbf{OSAlg}_{(\Sigma,E)}$ with φ satisfiable in A. Note that φ is *T-valid* iff $\neg\varphi$ is *T-unsatisfiable*.

$\Sigma = ((S, \leq), \Sigma)$ is a *subsignature* of $\Sigma' = ((S', \leq'), \Sigma')$, denoted $\Sigma \subseteq \Sigma'$, iff $(S, \leq) \subseteq (S', \leq')$ is a subposet inclusion, and $\Sigma \subseteq \Sigma'$. A *signature map* $H : \Sigma \to \Sigma'$ is a monotonic function $H : (S, \leq) \to (S', \leq')$ of the underlying posets of sorts together with a mapping $H : \Sigma \ni (f : s_1 \ldots s_n \to s) \mapsto (H(f) :$

[2] There is only an apparent lack of predicate symbols. To express a predicate $p(x_1{:}s_1, \ldots, x_n{:}s_n)$, add a new sort *Truth* with a constant tt, and with $\{Truth\}$ a separate connected component, and view p as a function symbol $p : s_1, \ldots, s_n \to Truth$. An atomic formula $p(t_1, \ldots, t_n)$ is then expressed as the equation $p(t_1, \ldots, t_n) = tt$.

$H(s_1)\dots H(s_n) \to H(s)) \in \Sigma'$. H induces a map $H : Form(\Sigma) \to Form(\Sigma')$. A signature inclusion $\Sigma \subseteq \Sigma'$ defines a signature map $\Sigma \hookrightarrow \Sigma' : f \mapsto f$.

A signature map $H : \Sigma \to \Sigma'$ induces a functor in the *opposite* direction $\_\,|_H$: $\mathbf{OSAlg}_{\Sigma'} \ni B \mapsto B\,|_H \in \mathbf{OSAlg}_{\Sigma}$, where the *H-reduct* $B\,|_H$ has: (i) for each $s \in S$, $(B\,|_H)_s = B_{H(s)}$; and (ii) for each $f : s_1 \dots s_n \to s$ in Σ, $(B\,|_H)_f = B_{H(f)}$. For $H : \Sigma \hookrightarrow \Sigma'$ a signature inclusion, $B\,|_H$ is denoted $B\,|_\Sigma$. For $B \in \mathbf{OSAlg}_{\Sigma'}$ and $\varphi \in Form(\Sigma)$ with $fvars(\varphi) = \emptyset$ we have [21]:

$$(\dagger) \qquad B \models H(\varphi) \;\Leftrightarrow\; B\,|_H \models \varphi.$$

3 Order-Sorted Rewriting and Equality

Given an OS signature $\Sigma = ((S, \leq), \Sigma)$, a *$\Sigma$-rewrite rule*[3] is a sequent $l \to r$ with $l, r \in T_\Sigma(X)_{[s]}$ for some $[s] \in \widehat{S}$. An *order-sorted term rewriting system* (OSTRS) is then a pair (Σ, R) with R a set of Σ-rewrite rules.

Since, as shown in the Introduction, replacement of equals for equals and standard rewriting break down in the order-sorted case, we should define rewriting deductions with an OSTRS not by means of the reflexive-transitive closure $\to_R^*$ of the rewrite relation $\to_R$, but by means of an *inference system* with two kinds of *sequents*: sequents $t \to t'$, where $t, t' \in T_\Sigma(X)_{[s]}$, $[s] \in \widehat{S}$, corresponding to *one-step* application of rules, and sequents $t \to^{\circledast} t'$, where $t, t' \in T_\Sigma(X)_{[s]}$, $[s] \in \widehat{S}$, corresponding to more complex rewriting deductions. The symbol $\to^{\circledast}$ is close enough to $\to^*$ to suggest that: (i) it plays a role similar to a reflexive transitive-closure in the unsorted case, but (ii) in general it is *different* from such a closure. For example, for Σ the signature in the Introduction and $R = \{a \to b, b \to c\}$, we can derive $f(a) \to^{\circledast} f(c)$, but there is no sequence of one-step rewrites from $f(a)$ to $f(c)$. We then define two kinds of *rewriting deductions*: $(\Sigma, R) \vdash t \to t'$ and $(\Sigma, R) \vdash t \to^{\circledast} t'$, as those sequents derivable from (Σ, R) by a finite application of the following inference rules, where σ denotes an S-sorted *substitution*, i.e., an S-sorted function $\sigma \in [X \to T_\Sigma(X)]$:

Reflexivity	$\dfrac{}{t \to^{\circledast} t}$
Subsumption	$\dfrac{t \to t'}{t \to^{\circledast} t'}$
Transitivity	$\dfrac{t \to^{\circledast} t' \qquad t' \to^{\circledast} t''}{t \to^{\circledast} t''}$
Congruence	$\dfrac{u_1 \to^{\circledast} u_1' \quad \dots \quad u_n \to^{\circledast} u_n'}{f(u_1, \dots, u_n) \to^{\circledast} f(u_1', \dots, u_n')}$ where $f(u_1, \dots, u_n), f(u_1', \dots, u_n') \in T_\Sigma(X)$
Replacement	$\dfrac{}{t\sigma \to t'\sigma}$ where $t \to t' \in R$

[3] For greater generality no restriction is placed on the variables of l and r.

The first three and the last inference rule are standard, but the **Congruence** rule is more subtle. We can better understand these rules by means of our running example (Σ, R). The sequent $f(a) \rightarrow^{\circledast} f(b)$ is *not* derivable: the attempt to obtain it by applying **Replacement** with rule $a \rightarrow b$, **Subsumption** to get $a \rightarrow^{\circledast} b$, and then **Congruence** fails, because of the side condition, since $f(b) \notin T_\Sigma(X)$. To see what *can* be derived, consider the derivation of the sequent $f(a) \rightarrow^{\circledast} f(c)$. Since we have rules $a \rightarrow b$ and $b \rightarrow c$, we can derive $a \rightarrow^{\circledast} c$ by two applications of **Replacement** followed by **Subsumption** and one application of **Transitivity**. Then **Congruence** gives us:

$$\frac{a \rightarrow^{\circledast} c}{f(a) \rightarrow^{\circledast} f(c)}$$

Note the interesting fact that $f(a)$ is typed with $f : A \rightarrow A$, and $f(c)$ is typed with $f : C \rightarrow C$. We can think of **Congruence** as a "tunneling rule." $f(a) \rightarrow^{\circledast} f(c)$ *cannot* be obtained by composing one-step rewrites: failed attempts such as that for deriving $f(a) \rightarrow^{\circledast} f(b)$ make it impossible; but we can "tunnel through" such failed attempts and obtain a more complex sequent like $f(a) \rightarrow^{\circledast} f(c)$ when the left- and right-hand sides are well-formed terms in $T_\Sigma(X)$.

The above inference system yields as a *special case* a sound and complete inference system for *order-sorted equational logic*: we just view an order-sorted equational theory (Σ, E) as the OSTRS $(\Sigma, R(E))$, where $R(E) = \{t \rightarrow t' \mid t = t' \in E \ \vee \ t' = t \in E\}$. That is, equality steps are viewed as either left-to-right or right-to-left rewrite steps. We then have:

Definition 3. *Given an order-sorted equational theory (Σ, E) with Σ sensible, its* equational deduction relation*, denoted $(\Sigma, E) \vdash u = v$, or just $E \vdash u = v$, is defined by the equivalence:*

$$(\Sigma, E) \vdash u = v \quad \Leftrightarrow \quad (\Sigma, R(E)) \vdash u \rightarrow^{\circledast} v.$$

Theorem 3 *(Soundness and Completeness) [21] Theorem 24. For Σ sensible and $E \cup \{u = v\}$ a set of Σ-equations we have the equivalence:*

$$(\Sigma, E) \vdash u = v \quad \Leftrightarrow \quad (\Sigma, E) \models u = v$$

The above theorem has as a corollary the construction of the *initial algebra* $T_{\Sigma/E}$ for the category $\mathbf{OSAlg}_{(\Sigma,E)}$ of (Σ, E)-algebras. Assuming Σ sensible, $T_{\Sigma/E}$, has an easy definition. Note that the relation $E \vdash u = v$ induces an equivalence relation $=_E$ on each set $T_{\Sigma,[s]}$, $[s] \in \widehat{S}$. We then define $T_{\Sigma/E,s'} = \{[t]_{=_E} \in T_{\Sigma,[s]}/=_E \mid [t]_{=_E} \cap T_{\Sigma,s'} \neq \emptyset\}$ for each $s' \in [s]$, and define each operation $f : s_1 \ldots s_n \rightarrow s \in \Sigma$ by the map $([t_1]_{=_E}, \ldots, [t_1]_{=_E}) \mapsto [f(t'_1, \ldots, t'_n)]_{=_E}$, where $t'_i \in [t_i]_{=_E} \cap T_{\Sigma,s_i}$, $1 \leq i \leq n$, showing it does not depend on the choice of t'_i's.

3.1 Kind-Complete OS-Rewriting and Equational Deduction

The order-sorted rewrite relation $t \rightarrow^{\circledast} t'$ is obviously quite impractical and hard to implement. For this reason, given an OSTRS (Σ, R) several conditions on

either Σ or R have been sought to be able to perform rewriting computations in essentially the standard and efficient way in which it is performed in an unsorted or many-sorted TRS. Two such conditions, going back to [14], are to either: (i) require that the rules R are *sort-decreasing*, i.e., for each $l \to r \in R$ and S-sorted substitution σ, if $l\sigma \in T_{\Sigma,s}$ then $r\sigma \in T_{\Sigma,s}$ (this can be checked by the method explained in [17]); or (ii) if R is not sort-decreasing, extend Σ with new "retract operators" $r_{s,s'} : s \to s'$, $s, s' \in [s]$, $s \not\leq s'$, to catch typing errors, add to R "error recovery" rules of the form $r_{s,s'}(x{:}s') \to x{:}s'$, and force sort-decreasingness of R by replacing each not sort-decreasing $u \to v \in R$ by suitable rules of the form $u\sigma \to r_{s,s'}(v\sigma)$, where σ may lower the sorts of some variables.

Conditions (i) or in its defect (ii) work and can be shown to be conservative in a certain sense [14]. However, they have serious limitations. Sort decreasingness is a strong condition that may be impossible to achieve for some OSTRS arising in practice; and if the solution with retracts is adopted, an unpleasant consequence is that we *change the models*, including the initial ones, since retracts add new operations and new error terms *to the original sorts*.

All these limitations can be avoided —while allowing rewriting with rules R and equational deduction with equations E to be performed in the *standard* way— by using a *faithful embedding* of order-sorted equational logic into *membership equational logic* (MEL) [3,21]. MEL introduces a typing distinction between *sorts* $s \in S$, which may be related by subsort relations just as in the order-sorted way, and the *kind* $\top_{[s]}$ associated to each connected component $[s] \in \widehat{S}$, which is above all sorts in $[s]$. An ill-formed term like $f(b)$ in the OS signature of the Introduction has no sort, but has kind $\top_{[B]}$. In this way, the earlier side condition in the **Congruence** rule in Sect. 3 can be avoided.

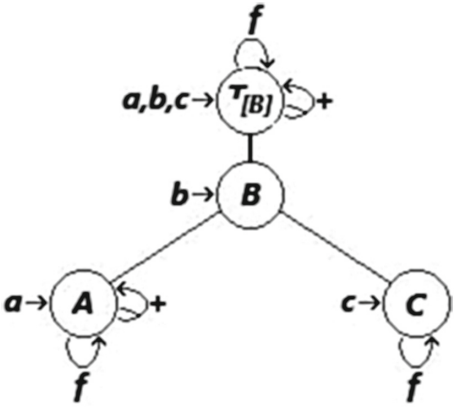
The faithfulness of this embedding of logics means in particular that *both* initial models and equational deduction are preserved ([21], Corollary 28). However: (i) the proof in [21] is model-theoretic; (ii) it focuses on the equational logic level, and does not deal with the more general rewriting logic level; and (iii) it assumes that the entire MEL framework is adopted. Can the essential advantages of this embedding be still obtained *while remaining at the order-sorted level*? The answer is *yes!* Since: (i) this solution plays a key role in the treatment of satisfiability for the theory of OS uninterpreted function symbols in Sect. 4, and (ii) having a much simpler theory of OS rewriting is useful in its own right, I give a detailed treatment of it below. The key idea is to use a signature transformation $\Sigma \mapsto \Sigma^{\Box}$ extending any OS signature Σ into one whose components have a top sort, understood as the kind of that component. The essential point is that $\Sigma^{\Box}$ belongs to a class of order-sorted signatures called *kind complete* where both rewriting and equational deduction can be performed in the standard way.

Definition 4. *An OS signature* $\Sigma = ((S, \leq), \Sigma)$ *is called* kind-complete *iff each connected component* $[s] \in \widehat{S}$ *has a top sort* $\top_{[s]}$*, called its* kind*, with* $\top_{[s]} \geq s'$ *for each* $s' \in [s]$*, and any non-empty subsort-polymorphic family* $f^{[s_1]\ldots[s_n]}_{[s]} \subseteq \Sigma$

includes the typing $f : \top_{[s_1]}, \ldots, \top_{[s_n]} \to \top_{[s]}$. *Note that any many-sorted* Σ — *and in particular any unsorted (i.e., single-sorted)* Σ — *is trivially kind-complete.*

Any OS signature Σ *can be extended to a kind-complete one by a transformation* $\Sigma \mapsto \Sigma^{\square}$. $\Sigma^{\square}$ *is constructed in two-steps: (i) we first associate to the order-sorted signature* $((S, \leq), \Sigma)$ *the many-sorted signature* $\widehat{\Sigma} = (\widehat{S}_{\top}, \widehat{\Sigma})$, *where* $\widehat{S}_{\top} = \{\top_{[s]} \mid [s] \in \widehat{S}\}$, *and with* $f : \top_{[s_1]} \ldots \top_{[s_n]} \to \top_{[s]} \in \widehat{\Sigma}$ *iff* $f^{[s_1]\ldots[s_n]}_{[s]} \neq \emptyset$; *and (ii) we then define* $\Sigma^{\square} = ((S \uplus \widehat{S}_{\top}, \leq_{\square}), \Sigma \uplus \widehat{\Sigma})$, *where* $\leq_{\square} \cap S^2 = \leq$, *and for each* $\top_{[s]} \in \widehat{S}_{\top}$ *we have* $s' <_{\square} \top_{[s]}$ *for each* $s' \in [s]$. *That is, we add* $\top_{[s]}$ *as a top sort above each* $s' \in [s]$ *and add the new typing* $f : \top_{[s_1]} \ldots \top_{[s_n]} \to \top_{[s]}$ *for each* $f^{[s_1]\ldots[s_n]}_{[s]} \neq \emptyset$.

For Σ the signature in the Introduction, $\Sigma^{\square}$ is as follows:



Instead, the many-sorted signature $\widehat{\Sigma}$ in this example happens to be unsorted, and is obtained by keeping only the sort $\top_{[B]}$ in the above figure, with the operations f and $+$ and constants a, b, c of of sort $\top_{[B]}$, and removing all other sorts and operations in the figure. In summary, $\Sigma^{\square}$ is the signature obtained by adding a new top sort $\top_{[s]}$ on top of each connected component $[s]$ and "lifting" to those top sorts all operations and constants, whereas $\widehat{\Sigma}$ is the many sorted signature obtained when we remove from $\Sigma^{\square}$ all sorts except the newly added top sorts of the form $\top_{[s]}$ for each $[s]$.

We then have subsignature inclusions: $\Sigma \subseteq \Sigma^{\square}$ and $\widehat{\Sigma} \subseteq \Sigma^{\square}$. Note that, by construction, if Σ is sensible, both $\widehat{\Sigma}$ and $\Sigma^{\square}$ are also sensible; and that the initial algebra $T_{\Sigma^{\square}}$ is *preserved by reducts*, i.e., we have:

$$T_{\Sigma^{\square}}|_{\Sigma} = T_{\Sigma} \quad \textit{and} \quad T_{\Sigma^{\square}}|_{\widehat{\Sigma}} = T_{\widehat{\Sigma}}.$$

For kind-complete signatures, rewriting, and in particular equational deduction, can be performed in the standard, sorted way. Recall the usual notation to denote term positions, subterms, decompositions and term replacement from [6]: (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, (iii)

$t = t[t|_p]_p$ denotes a *decomposition* of t into a context $t[]_p$ and its subterm $t|_p$, and (iv) $t[u]_p$ denotes the result of *replacing* subterm $t|_p$ at position p by u.

Definition 5. *Let (Σ, R) be an OSTRS with Σ sensible and kind-complete. The* one-step *R-rewrite relation $u \to_R v$ holds between $u, v \in T_\Sigma(X)_{[s]}$, $[s] \in \widehat{S}$, iff there is a rewrite rule $t \to t' \in R$, a substitution $\sigma \in [X \to T_\Sigma(X)]$, and a term position p in u such that $u = u[t\sigma]_p$ and $v = u[t'\sigma]_p$.*

We denote by $\to_R^+$ the transitive closure of $\to_R$, and by $\to_R^*$ the reflexive-transitive closure of $\to_R$, and write $(\Sigma, R) \vdash u \to_R^* v$ to make Σ explicit.

(Σ, R) is called terminating *iff $\to_R$ is a well-founded relation; and is called* confluent *iff whenever $t \to_R^* u$ and $t \to_R^* v$ there exists w such that $u \to_R^* w$ and $v \to_R^* w$. (Σ, R) is called* convergent *iff it is both confluent and terminating. If (Σ, R) is convergent, each Σ-term t rewrites by some $t \to_R^* t!_R$ to a unique term $t!_R$, called its R-*canonical form*, that cannot be further rewritten.*

When Σ is kind-complete, if $u \in T_\Sigma(X)_{[s]}$, $t \to t' \in R$, and $u = u[t\sigma]_p \in T_\Sigma(X)_{[s]}$, then we always have $u[t'\sigma]_p \in T_\Sigma(X)_{[s]}$. That is, $\to_R$ *never produces ill-formed terms*, so that in the above definition of $\to_R$ the requirement the $v \in T_\Sigma(X)_{[s]}$ is unnecessary and does not have to be checked. Indeed, for kind-complete signatures order-sorted rewriting *becomes standard sorted rewriting*:

Lemma 1. *Let (Σ, R) be an OSTRS with Σ sensible and kind-complete. Then we have the equivalence:*

$$(\Sigma, R) \vdash u \to^{\circledast} v \quad \Leftrightarrow \quad (\Sigma, R) \vdash u \to_R^* v.$$

Corollary 1. *Let Σ be a sensible and kind-complete OS signature, and $E \cup \{u = v\}$ a set of Σ-equations. Then we have the equivalence:*

$$(\Sigma, E) \vdash u = v \quad \Leftrightarrow \quad (\Sigma, R(E)) \vdash u \to_{R(E)}^* v.$$

3.2 Conservativity Results

The whole point of the signature transformation $\Sigma \mapsto \Sigma^\Box$ is to replace complex deductions of the form $(\Sigma, R) \vdash u \to^{\circledast} v$ by simple rewrite sequences $u \to_R^* v$ in the *extended* OSTRS $(\Sigma^\Box, R)$. But is this sound?

Theorem 4. *Let (Σ, R) be an OSTRS with Σ sensible. Then for any $u, v \in T_\Sigma(X)_{[s]}$, $[s] \in \widehat{S}$ we have the equivalence:*

$$(\Sigma, R) \vdash u \to^{\circledast} v \quad \Leftrightarrow \quad (\Sigma^\Box, R) \vdash u \to_R^* v.$$

Corollary 2. *Let Σ be a sensible OS signature and $E \cup \{u = v\}$ a set of Σ-equations. Then we have the equivalences:*

$$(\Sigma, E) \vdash u = v \quad \Leftrightarrow \quad (\Sigma^\Box, E) \vdash u = v \quad \Leftrightarrow \quad (\Sigma^\Box, R(E)) \vdash u \to_{R(E)}^* v.$$

Since, besides the subsignature inclusion $\Sigma \subseteq \Sigma^\Box$, we also have the inclusion $\widehat{\Sigma} \subseteq \Sigma^\Box$, we have a further conservativity result:

Lemma 2. *Let Σ be a sensible OS signature and $(\widehat{\Sigma}, R)$ a many-sorted TRS. Then for any $u, v \in T_{\widehat{\Sigma}}(X)_{\top_{[s]}}$, $\top_{[s]} \in \widehat{S}_{\top}$, where $X = \{X_{\top[s]}\}_{\top_{[s]} \in \widehat{S}_{\top}}$, we have $(\widehat{\Sigma}, R) \vdash u \rightarrow_R^* v$ iff $(\Sigma^{\Box}, R) \vdash u \rightarrow_R^* v$. As an immediate consequence, for $E \cup \{u = v\}$ a set of $\widehat{\Sigma}$-equations, we have the equivalence:*

$$(\widehat{\Sigma}, E) \vdash u = v \quad \Leftrightarrow \quad (\Sigma^{\Box}, E) \vdash u = v.$$

4 Order-Sorted $(\Sigma, \emptyset)$-QF-Satisfiability

In theorem proving the theory $(\Sigma, \emptyset)$, whose category of algebras is $\mathbf{OSAlg}_{\Sigma}$, is called the theory of *uninterpreted function symbols* Σ. As remarked in Definition 1, a *many-sorted* signature Σ is a special case of an order-sorted signature, and an *unsorted* signature is a many-sorted signature where $S = \{U\}$ is a singleton set. Let $QFForm(\Sigma) \subseteq Form(\Sigma)$ denote the set of *quantifier-free* Σ-formulas, i.e., formulas with no quantifiers. When Σ is unsorted, $(\Sigma, \emptyset)$-*QF-satisfiability*, i.e., $(\Sigma, \emptyset)$-satisfiability for any $\varphi \in QFForm(\Sigma)$ is *decidable* [1]. The goal of this section is to show that the same holds for any sensible OS signature Σ by a *reduction* method. This can be done by two reductions. The first reduces this decidability problem to that of the *OS word problem*, which is the problem of whether, given a sensible OS signature Σ and a finite set $E \cup \{u = v\}$ of *ground* Σ-equations, $E \vdash u = v$ holds or not. The desired first reduction is as follows:

Theorem 5. *$(\Sigma, \emptyset)$-QF-satisfiability is decidable for any sensible order-sorted signature Σ iff the OS word problem is decidable.*

The proof follows from the more general Theorem 7 in Sect. 5, which deals with the OS word problem *modulo* equations B. The theorem's algorithmic content mirrors its proof: $\varphi = \bigvee_{1 \leq i \leq n} (\bigwedge E_i \wedge \bigwedge D_i)$ in DNF with the E_i equalities and the D_i disequalities is satisfiable iff, when we view the variables in φ as fresh new constants C, there is an i, $1 \leq i \leq n$, such that $E_i \nvdash u = v$ for each $u \neq v \in D_i$. Furthermore, $\bigwedge E_i \wedge \bigwedge D_i$ is satisfiable iff $T_{\Sigma(C)/E_i} \models \bigwedge E_i \wedge \bigwedge D_i$.

The second reduction is from the OS word problem to the *unsorted* word problem. This is broken into *two* reductions: (i) of the many-sorted word problem to the unsorted word problem in Sect. 4.1, and (ii) of the OS word problem to the many-sorted word problem in Sect. 4.2.

For Σ *unsorted* and $E \cup \{u = v\}$ a finite set of ground Σ-equations it is well-known that the word problem $E \vdash u = v$ can be decided by a *congruence closure* algorithm [7,24,27]. What the various such algorithms have in common is that they are all instances (by applying difference strategies) of the same *abstract congruence closure* algorithm in the sense of [2], which is summarized below.

4.1 Abstract Congruence Closure

What the abstract congruence closure algorithm in [2] captures is what all concrete congruence closure algorithms have in common: they all are efficient, specialized *ground Knuth-Bendix* completion algorithms [2,12,16,18]: they all begin

with a set E of ground equations, and return a set R of *convergent* ground rewrite rules R equivalent to E (on a possibly extended signature). We can then decide the word problem $E \vdash u = v$ by checking the syntactic equality $u!_R = v!_R$.

The key notion of *abstract congruence closure* in [2] is then as follows:

Definition 6. *[2] For Σ an unsorted signature and E a finite set of ground Σ-equations, an* abstract congruence closure *for E is a set R of ground convergent $\Sigma(K)$-rewrite rules, where K is a finite set of new constants, such that: (i) they are either of the form $c \to c'$, with $c, c' \in K$, or of the form $f(c_1, \ldots, c_n) \to c$, with $c_1, \ldots, c_n, c \in K$, $f \in \Sigma$ with $n \geq 0$ arguments; (ii) for each $c \in K$ there is a ground Σ-term t such that $t!_R = c!_R$; and (iii) for any ground Σ-equation $u = v$ we have $E \vdash u = v$ iff we have the syntactic equality $u!_R = v!_R$.*

The paper [2] then gives an *abstract congruence closure algorithm* described by six inference rules, with an optional seventh, such that: (i) takes as input a triple $(\emptyset, E, \emptyset)$ with E is a set of ground Σ-equations; (ii) operates on triples of the form (K', E', R') with E' (resp. R') the current $\Sigma(K')$-equations (resp. $\Sigma(K')$-rules); and (iii) terminates with a triple of the form $(K, \emptyset, R)$ such that R is a congruence closure for E. The name *abstract congruence closure* is well-deserved: the algorithms in [7,24,27], and two other ones, are all shown to be *instantiations* of the abstract algorithm by applying the inference rules with different *strategies*, so that both the operation of each algorithm and its actual complexity are faithfully captured by the corresponding instantiation [2].

We need to decide *the many-sorted* word problem as a step for deciding the more general order-sorted one. But the many-sorted word problem can be easily *reduced* to the unsorted one by means of the signature transformation $\Sigma \ni (f : s_1 \ldots s_n \to s) \mapsto (f : U \,.^n.\, U \to U) \in \Sigma^u$, where $\Sigma = (S, \Sigma)$ is a many-sorted signature. Then all boils down to the following lemma:

Lemma 3. *For Σ a sensible many-sorted signature and E a set of* regular *Σ-equations —i.e., t and t' have the same variables for each $t = t' \in E$— we have $(\Sigma, E) \vdash u = v$ iff $(\Sigma^u, E^u) \vdash (u = v)^u$, where for any Σ-equation $t = t'$, $(t = t')^u$ leaves the terms unchanged but regards all variables as unsorted.*

This lemma has a very practical consequence: we can use an unsorted congruence closure algorithm to solve the many-sorted word problem *at no extra cost*: no changes are needed either to the input E or to the unsorted algorithm.

4.2 Deciding OS $(\Sigma, \emptyset)$-QF-Satisfiability

For any sensible OS signature Σ we have reduced the decidability of the $(\Sigma, \emptyset)$-QF-satisfiability problem to that of the OS word problem in Theorem 5. And in Lemma 3 we have reduced the many-sorted word problem to the unsorted word problem, which is decidable by a congruence closure algorithm. To prove the decidability of the OS $(\Sigma, \emptyset)$-QF-satisfiability problem and obtain a correct algorithm for it we just need to reduce the OS word problem to the many-sorted word problem. For this, the conservativity results in Sect. 3.2 are crucial:

Theorem 6. *Let Σ be a sensible OS signature and $E \cup \{u = v\}$ a set of ground Σ-equations. Then we have the equivalence:*

$$(\Sigma, E) \vdash u = v \quad \Leftrightarrow \quad (\widehat{\Sigma}, E) \vdash u = v.$$

The decidability of the OS $(\Sigma, \emptyset)$-QF-satisfiability problem goes back to [11]; but the reduction achieved by Theorem 5, Lemma 3 and Theorem 6 yields a new, very simple algorithm. Either by already having φ in DNF or by using a DPLL$(\Sigma, \emptyset)$ solver, deciding the satisfiability of φ boils down to finding a satisfiable conjunction $\bigwedge E \wedge \bigwedge D$, with E (resp. D) a finite sets of equations (resp. disquations), which can be viewed as a *ground* $\Sigma(C)$-formula by adding $C = \mathit{fvars}(\varphi)$ as *new constants*. Then, satisfiability of $\bigwedge E \wedge \bigwedge D$ is decided by:

1. regarding at no cost $\bigwedge E \wedge \bigwedge D$ as a ground $\Sigma(C)^u$-formula,
2. computing a congruence closure R for E in the usual way [7,24,27], and
3. checking the syntactic inequality $u!_R \not\equiv v!_R$ for each $u \neq v \in D$.

Therefore we can *reuse* the same algorithms and tools used in the *unsorted* case *at no extra cost*: the input to such algorithms and the algorithms or tools themselves need no changes, and the complexity is that of the unsorted case.

5 Order-Sorted (Σ, AC_Δ)-QF-Satisfiability

Let Σ be a sensible OS signature with $\Delta \subseteq \Sigma$ made exclusively of binary function symbols, say, $g, h, \ldots$, each of the form $g : s\,s \to s$ for some sorts $s \in S$, and with any typing of any such g in Σ necessarily a typing in Δ, i.e., Δ and $(\Sigma - \Delta)$ share no symbols. Assume that each non-empty subsort-polymorphic family $g^{[s]\,[s]}_{[s]} \subseteq \Delta$ has always a biggest possible typing $g : s_g\,s_g \to s_g$ such that for any other typing $g : s\,s \to s$ in $g^{[s]\,[s]}_{[s]}$ we have $s \leq s_g$. The equations: $AC_g = \{g(x, y) = g(y, x), g(x, g(y, z)) = g(g(x, y), z)\}$, with x, y, z of sort s_g, express the *associativity-commutativity* (AC) of the subsort-polymorphic family $g^{[s]\,[s]}_{[s]}$. We require that the axioms AC_g are *sort-preserving*, that is, that for each S-sorted substitution σ and each sort $s \in S$ we have: $g(x, y)\sigma \in T_\Sigma(X)_s \;\Leftrightarrow\; g(y, x)\sigma \in T_\Sigma(X)_s$, and $g(x, g(y, z))\sigma \in T_\Sigma(X)_s \;\Leftrightarrow\; g(g(x, y), z)\sigma \in T_\Sigma(X)_s$, which can be easily checked by the method explained in [17]. Let AC_Δ denote the set $AC_\Delta = \bigcup_{g \in \Delta} AC_g$ making all symbols in Δ AC. Call (Σ, AC_Δ) the OS theory of Σ *uninterpreted function symbols* Σ *modulo* AC_Δ. When $\Sigma = \Delta$ is unsorted and has a single symbol $+$, this is called the *theory of commutative semigroups*.

We can generalize the above setting by replacing (Δ, AC_Δ) by any OS theory (Δ, B) with Δ sensible and considering any sensible supersignature $\Sigma \supseteq \Delta$ with Δ and $\Sigma - \Delta$ not sharing any symbols. Call (Σ, B) the theory of *uninterpreted function symbols* Σ *modulo* B. We can then reduce the decidability of the (Σ, B)-QF-satisfiabilty problem to that of the *OS word problem modulo* B, defined as the problem of whether given any $\Sigma \supseteq \Delta$ as above, and a set $E \cup \{u = v\}$ of ground Σ-equations, $E \cup B \vdash u = v$ holds or not. The reduction is as follows:

Theorem 7. *For any (Δ, B) and $\Sigma \supseteq \Delta$ as above, (Σ, B)-QF-satisfiability is decidable iff the OS word problem modulo B is decidable.*

For $\Sigma \supseteq \Delta$ unsorted, there are AC *congruence closure* algorithms for the theory (Σ, AC_Δ) [2,19,22] that decide the word problem modulo AC_Δ and therefore, by above Theorem 7, the unsorted (Σ, AC_Δ)-QF-satisfiability problem. In the spirit of Sect. 4, the main goal of this section is to *reduce* the decidability of the OS (Σ, AC_Δ)-QF-satisfiability problem to that of its unsorted version, and to furthermore *reuse* the *same* unsorted AC congruence closure algorithms in [2,19,22] to decide *at no extra cost* and with the *same complexity* the OS (Σ, AC_Δ)-QF-satisfiability problem.

The decidability of OS (Σ, AC_Δ)-QF-satisfiability has already been reduced to that of the OS word problem modulo AC_Δ, now we just need to reduce the OS word problem modulo AC_Δ to the unsorted word problem modulo AC_{Δ^u}.

This is achieved in two steps. First, we reduce the many-sorted word problem modulo $AC_{\widehat{\Delta}}$ to the unsorted word problem modulo AC_{Δ^u} using the $\widehat{\Sigma} \mapsto \Sigma^u$ transformation of Sect. 4.1. This first reduction is easy: the equations $AC_{\widehat{\Delta}}$ are *regular*. Therefore, if $E \cup \{u = v\}$ is a finite set of ground many-sorted $\widehat{\Sigma}$-equations, the equations $E \cup AC_{\widehat{\Delta}}$ are also regular and the conditions of Lemma 3 apply. We then reduce the OS word problem modulo AC_Δ to the many-sorted word problem modulo $AC_{\widehat{\Delta}}$. The $\widehat{\Delta}$-equations $AC_{\widehat{\Delta}}$ are obtained from the OS Δ-equations in AC_Δ by replacing each variable $x{:}s$ by the variable $x{:}\top_{[s]}$. That is, for $E \cup \{u = v\}$ a finite set of *ground* Σ-equations must show the equivalence:

$$(\Sigma, E \cup AC_\Delta) \vdash u = v \quad \Leftrightarrow \quad (\widehat{\Sigma}, E \cup AC_{\widehat{\Delta}}) \vdash u = v$$

which, by Corollary 2, reduces to proving the equivalence:

$$(\Sigma^\square, E \cup AC_\Delta) \vdash u = v \quad \Leftrightarrow \quad (\widehat{\Sigma}, E \cup AC_{\widehat{\Delta}}) \vdash u = v$$

which, by Lemma 2, follows as a special case from the more general theorem:

Theorem 8. *Let $\Sigma \supseteq \Delta$ be a sensible OS supersignature, R a set of Σ-rewrite rules, and $u, v \in T_\Sigma(X)$. Then we have the equivalence:*

$$(\Sigma^\square, R \cup R(AC_\Delta)) \vdash u \rightarrow^*_{R \cup R(AC_\Delta)} v \;\Leftrightarrow\; (\Sigma^\square, R \cup R(AC_{\widehat{\Delta}})) \vdash u \rightarrow^*_{R \cup R(AC_{\widehat{\Delta}})} v.$$

6 Related Work and Conclusions

[11] presents the only *order-sorted* congruence closure algorithm I am aware of. It provides a good solution under some extra assumptions on Σ, but it requires a quite complex congruence generation method and has worse complexity, $O(n^2)$, than the best $O(n\ log(n))$ unsorted algorithms. The papers [2,12,16,19,22] present the view of congruence closure as completion. In particular, I have used *abstract congruence closure* [2] and *AC-congruence closure* [2,19,22]. The modular combination of congruence closure, AC congruence, and

polynomial ring congruence closure algorithms for different symbols and its relation to the Nelson-Oppen combination method [23,25] is studied in [31]. Likewise, the combination of AC congruence closure with other satisfiability algorithms using the Shostak combination method [28] is studied in [5] The first general study I know of satisfiability modulo theories in an order-sorted setting is [30].

The above-mentioned work has influenced and motivated the present one. The good news is that we get all the benefits of order-sorted $(\Sigma, \emptyset)$- and (Σ, AC_Δ)-satisfiability *for free*, with no added computational cost and being able to reuse unsorted tools. At a more theoretical level, the order-sorted rewriting and equality results presented here are also good news and belong to the foundations of such an area. Future work will focus on exploiting these results at the tool level.

Acknowledgements. Partially supported by NSF Grant CNS 13-19109. I thank Maria Paola Bonacina for suggested improvements.

References

1. Ackermann, W.: Solvable Cases of the Decision Problem. North-Holland Publishing Company, Amsterdam (1954)
2. Bachmair, L., Tiwari, A., Vigneron, L.: Abstract congruence closure. J. Autom. Reasoning **31**(2), 129–168 (2003)
3. Bouhoula, A., Jouannaud, J.P., Meseguer, J.: Specification and proof in membership equational logic. Theoret. Comput. Sci. **236**, 35–132 (2000)
4. Clavel, M., Durán, F., Eker, S., Meseguer, J., Lincoln, P., Martí-Oliet, N., Talcott, C.: All About Maude. LNCS, vol. 4350. Springer, Heidelberg (2007)
5. Conchon, S., Contejean, E., Iguernelala, M.: Canonized rewriting and ground AC completion modulo Shostak theories : design and implementation. Logical Methods Comput. Sci. **8**(3), 1–29 (2012)
6. Dershowitz, N., Jouannaud, J.P.: Rewrite systems. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. B, pp. 243–320. North-Holland Publishing Company, Amsterdam (1990)
7. Downey, P.J., Sethi, R., Tarjan, R.E.: Variations on the common subexpressions problem. J. ACM **27**(4), 758–771 (1980)
8. Frisch, A.M.: The substitutional framework for sorted deduction: fundamental results on hybrid reasoning. Artif. Intell. **49**(1–3), 161–198 (1991)
9. Futatsugi, K., Diaconescu, R.: CafeOBJ Report. World Scientific, Singapore (1998)
10. Futatsugi, K., Goguen, J., Jouannaud, J.P., Meseguer, J.: Principles of OBJ2. In: Proceedings of POPL 1985, pp. 52–66. ACM (1985)
11. Gallier, J., Isakowitz, T.: Order-sorted congruence closure. Technical report CIS-686, UPenn (1988). http://repository.upenn.edu/cis_reports/686
12. Gallier, J.H., Narendran, P., Plaisted, D.A., Raatz, S., Snyder, W.: An algorithm for finding canonical sets of ground rewrite rules in polynomial time. J. ACM **40**(1), 1–16 (1993)
13. Gnaedig, I., Kirchner, C., Kirchner, H.: Equational completion in order-sorted algebras. Theoret. Comput. Sci. **72**(2–3), 169–202 (1990)

14. Goguen, J., Jouannaud, J.P., Meseguer, J.: Operational semantics of order-sorted algebra. In: Brauer, W. (ed.) Automata, Languages and Programming. LNCS, vol. 194, pp. 221–231. Springer, Heidelberg (1985)
15. Goguen, J., Meseguer, J.: Order-sorted algebra I. Theoret. Comput. Sci. **105**, 217–273 (1992)
16. Kapur, D.: Shostak's congruence closure as completion. In: Comon, H. (ed.) RTA 1997. LNCS, vol. 1232, pp. 23–37. Springer, Heidelberg (1997)
17. Kirchner, C., Kirchner, H., Meseguer, J.: Operational semantics of OBJ3. In: Lepistö, T., Salomaa, A. (eds.) Automata, Languages and Programming. LNCS, vol. 317, pp. 287–301. Springer, Heidelberg (1988)
18. Knuth, D., Bendix, P.: Simple word problems in universal algebra. In: Leech, J. (ed.) Computational Problems in Abstract Algebra. Pergamon Press, Oxford (1970)
19. Marché, C.: On ground AC-completion. In: Book, R.V. (ed.) RTA 1991. LNCS, vol. 488, pp. 411–422. Springer, Heidelberg (1991)
20. Meseguer, J.: Order-sorted rewriting and congruence closure. Technical report, C.S. Department, University of Illinois at Urbana-Champaign, June 2015. http://hdl.handle.net/2142/78008
21. Meseguer, J.: Membership algebra as a logical framework for equational specification. In: Parisi-Presicce, F. (ed.) WADT 1997. LNCS, vol. 1376, pp. 18–61. Springer, Heidelberg (1998)
22. Narendran, P., Rusinowitch, M.: Any ground associative-commutative theory has a finite canonical system. In: Book, R.V. (ed.) RTA 1991. LNCS, vol. 488, pp. 423–434. Springer, Heidelberg (1991)
23. Nelson, G., Oppen, D.C.: Simplification by cooperating decision procedures. ACM Trans. Program. Lang. Syst. **1**(2), 245–257 (1979)
24. Nelson, G., Oppen, D.C.: Fast decision procedures based on congruence closure. J. ACM **27**(2), 356–364 (1980)
25. Oppen, D.C.: Complexity, convexity and combinations of theories. Theoret. Comput. Sci. **12**, 291–302 (1980)
26. Schmidt-Schauss, M.: Computational Aspects of Order-Sorted Logic with Term Declarations. LNCS (LNAI), vol. 395. Springer, Heidelberg (1989)
27. Shostak, R.E.: An algorithm for reasoning about equality. Commun. ACM **21**(7), 583–585 (1978)
28. Shostak, R.E.: Deciding combinations of theories. J. ACM **31**(1), 1–12 (1984)
29. Smolka, G., Aït-Kaci, H.: Inheritance hierarchies: semantics and unification. J. Symb. Comput. **7**(3/4), 343–370 (1989)
30. Tinelli, C., Zarba, C.G.: Combining decision procedures for sorted theories. In: Alferes, J.J., Leite, J. (eds.) JELIA 2004. LNCS (LNAI), vol. 3229, pp. 641–653. Springer, Heidelberg (2004)
31. Tiwari, A.: Combining equational reasoning. In: Ghilardi, S., Sebastiani, R. (eds.) FroCoS 2009. LNCS, vol. 5749, pp. 68–83. Springer, Heidelberg (2009)
32. Walther, C.: A mechanical solution of Schubert's steamroller by many-sorted resolution. Artif. Intell. **26**(2), 217–224 (1985)