Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 290 (2012) 37–50

www.elsevier.com/locate/entcs

Order-sorted Equational Unification Revisited

Joe Hendrix[1,2]

Galois, Inc
Portland, OR 97204, USA

José Meseguer[1,3]

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Abstract

This paper presents a rule-based algorithm for performing order-sorted E-unification using an unsorted E-unification decision procedure under assumptions about E that are commonly satisfied in practice. We have implemented this algorithm in Maude for use with the Maude-NRL protocol analyzer and have used CiME for unsorted E-unification for E any set of AC and ACU axioms. In many examples of interest, using order-sorted unification over unsorted unification is able to reduce the total number of unifiers considered, and dramatically improve the performance of the Maude-NRL tool.

Keywords: Order-sorted unification, rule-based programming.

1 Introduction

Unification is a fundamental operation in many applications. For example, in solving reachability problems using narrowing, the unification procedure is called many times to unify terms representing reachable states against the left-hand sides of rewrite rules. This process is computationally expensive and often generates a large number of different terms — many of which may represent states that do not correspond to legal states. In order to avoid this problem tools such as the Maude-NRL protocol analyzer [5,6] use order-sorted algebras and rely on the sorts to only consider well-formed terms.

We present an algorithm which can use a procedure for *unsorted* E-unification to perform *order-sorted* E-unification under conditions general enough to cover many

[1] The research has been supported in part by ONR grant N00014-02-1-0715 and NSF grant CNS-07-16638.

[2] Email: jhendrix@galois.com

[3] Email: meseguer@cs.uiuc.edu

1571-0661
doi:10.1016/j.entcs.2012.11.010

practical applications. This algorithm solves a key challenge faced by the Maude-NRL protocol analyzer — most existing unification tools only support unsorted unification and ignore the sort information. Since equational unification procedures are often quite complex, it requires significantly less work to use an existing unification tool rather that writing an order-sorted equational unification procedure from scratch.

The order-sorted unification algorithm we present in this work can be naturally described by a terminating and confluent set of rewrite rules which compute order-sorted unifiers $\theta_1, \ldots, \theta_n$ for each unsorted unifier $\overline{\theta}$ returned by the unsorted unification procedure. We have implemented the algorithm in Maude, and have used CiME as the unsorted equational unification procedure. Our experimental results so far have shown that, although technically there may be many order-sorted unifiers for each unsorted unifier, this is rarely the case in practice. In fact, in practice there are usually fewer order-sorted unifiers than unsorted unifiers, and the use of order-sorted unification is essential for both correctness and performance, that is, so that the terms explored are always well-formed terms, and to ensure that the Maude-NRL analyzer is capable of handling real problems.

Our idea is not new, and was presented in [12] and more recently without a proof of correctness in [5]. However, after implementing these ideas in the Maude-NRL protocol analyzer, we felt that a new paper presenting the basic ideas was in order for several reasons:

- Our experience with the Maude-NRL protocol analyzer so far has suggested that for theories with AC operators, for practical protocol verification tools based on narrowing it is essential to use the sort information during unification. However, most existing unification procedures only perform unsorted E-unification and do not support sorts and subsorts. By using the techniques described in this work, one can obtain an order-sorted E-unification procedure from an unsorted one with very little effort for many equational theories.
- The algorithm in [12] was buried in a function's definition appearing in the proof of Theorem 34 in [12]. In this paper, we present a simple rule-based algorithm which is almost directly implementable in Maude. The algorithm only consists of three confluent and terminating rewrite rules, and it should be easily possible to compose these rules with inference steps in a modular way in other reasoning tools using unification.
- Perhaps most important from a technical perspective, the correctness results in [12] imposed unnecessarily strong technical conditions which excluded the majority of E-unification problems when E contains collapsing equations like idempotence $x + x = x$ and identity $x + 0 = x$. As identity was important for the Maude-NRL protocol analyzer and idempotence is a common axiom in many E-unification algorithms, in this paper we prove the correctness results under weaker assumptions about the equational theory and some technical assumptions about the unification engine. The assumptions about the unification engine should be satisfied in practice. Additionally, we show specifically how the algorithm can be

used in Maude for equational theories with any combination of free, commutative, AC, and ACU symbols.

This paper is organized as follows. In Section 2, we review basic definitions of order-sorted algebra and unification. In Section 3, we present our algorithm to compute order-sorted unifiers from unsorted unifiers. In Section 4, we illustrate how it can be used for AC and ACU order-sorted unification in Maude and, in Section 5 we prove its correctness. Finally, in Section 6, we discuss related work and suggest directions for future research.

2 Preliminaries

2.1 Order Sorted Algebra

An *order-sorted signature* $\Sigma = (S, F, \leq)$ consists of a set of *sorts* S, a family of *operators* $F = \{ F_{w,s} \}_{(w,s)\in S^* \times S}$, and a partial order $\leq \subseteq S \times S$ called the *subsort ordering*. We let $X = \{ X_s \}_{s\in S}$ denote a fixed S-sorted family of infinite sets of *variables* that are both pairwise disjoint for different sorts, i.e., $X_s \cap X_{s'} = \varnothing$ for distinct $s, s' \in S$, and disjoint from the operators F. As a notational convenience, we write x_s to denote that $x \in X_s$ when the variables X are clear from the context. The Σ-*terms* with variables X are members of the family $T_\Sigma(X) = \{ T_\Sigma(X)_s \}_{s\in S}$, where $T_{\Sigma,s}(X)$ denotes the Σ-terms with any sort $s' \leq s$. An *order-sorted theory* $\mathcal{E} = (\Sigma, E)$ consists of an order-sorted signature Σ and a finite set E of equations $l = r$ where $l, r \in T_{\Sigma,s}(X)$ for some sort $s \in S$. An *order-sorted substitution* is a function $\theta : Y \to T_\Sigma(X)$ with Y a finite subset of X, and for each variable $x_s \in Y$, $x_s\theta \in T_\Sigma(X)_s$. We let $\mathrm{rvars}(\theta)$ denote the variables occurring in a terms in the codomain of θ, i.e., $\mathrm{rvars}(\theta) = \bigcup_{x\in Y} \mathrm{vars}(x\theta)$. Given substitutions $\theta_1, \theta_2 : Y \to T_\Sigma(X)$, we write $\theta_1 =_\mathcal{E} \theta_2$ if $x\theta_1 =_\mathcal{E} x\theta_2$ for all $x \in Y$, and we write $\theta_1 \geq_\mathcal{E} \theta_2$ if there is a substitution $\psi : \mathrm{rvars}(\theta_1) \to T_\Sigma(X)$ such that $\theta_1\psi =_\mathcal{E} \theta_2$. For an equational theory $\mathcal{E} = (\Sigma, E)$, we define the relation $=_\mathcal{E} \subseteq T_\Sigma(X) \times T_\Sigma(X)$ as the least equivalence relation defined by the logical equivalence $t =_\mathcal{E} t' \iff \mathcal{E} \vdash (\forall X)\, t = t'$ where X is our fixed set of variables and $\vdash$ is the order-sorted deduction relation [8,11]. For each order-sorted theory $\mathcal{E} = (\Sigma, E)$ with $\Sigma = (S, F, \leq)$, there is an *underlying unsorted theory* $\overline{\mathcal{E}} = (\overline{\Sigma}, E)$ over variables $\overline{X} = \bigcup_{s\in S} X_s$ such that $\overline{\Sigma}$ is a ranked alphabet containing an operator f with arity n iff there is an operator $f \in F_{s_1 \ldots s_n, s}$ for some sorts $s_1, \ldots, s_n, s \in S$. Observe that a Σ-equation $l = r \in E$ is always a $\overline{\Sigma}$-equation.

2.2 Order-sorted Equational Unification

For a fixed order-sorted theory $\mathcal{E} = (\Sigma, E)$ with $\Sigma = (S, F, \leq)$, we define an order-sorted *unification problem* to be a finite conjunctive set Γ of Σ-equations $t = u$ where t and u are terms in $T_\Sigma(X)$ whose sorts belong to the same connected component in $(S, \leq)$. A $\mathcal{E}$-*unifier* for Γ is an order-sorted substitution $\theta : \mathrm{vars}(\Gamma) \to T_\Sigma(X)$ such that $t\theta =_\mathcal{E} u\theta$ for each equation $t = u \in \Gamma$. We denote the set of $\mathcal{E}$-unifiers for Γ by $\mathrm{Un}_\mathcal{E}(\Gamma)$, and we let $\mathrm{Un}_\Sigma(\Gamma)$ denote the syntactic unifiers for Γ, i.e., $\mathrm{Un}_\Sigma(\Gamma) = \mathrm{Un}_{(\Sigma,\varnothing)}(\Gamma)$. A set $S \subseteq \mathrm{Un}_\mathcal{E}(\Gamma)$ of $\mathcal{E}$-unifiers of Γ is *complete* if for all unifiers

$\psi \in \mathrm{Un}_{\mathcal{E}}(\Gamma)$, there is a unifier $\theta \in S$ such that $\theta \geq_{\mathcal{E}} \psi$. A set of $\mathcal{E}$-unifiers S is *most-general* if for distinct substitutions $\theta_1, \theta_2 \in S$, $\theta_1 \not\geq_{\mathcal{E}} \theta_2$. A given theory $\mathcal{E}$ has a *finitary* unification problem if there is a complete finite set of $\mathcal{E}$-unifiers S for each unification problem Γ.

3 Order-sorted Unification

Our main goal in this work is to develop a clear rule-based algorithm for solving order-sorted $\mathcal{E}$-unification problems using an unsorted $\overline{\mathcal{E}}$-unification procedure. In order to show that the rule-based algorithm returns a complete set of most-general unifiers, there are some technical requirements placed on the order-sorted theory $\mathcal{E}$ as well as on the most-general unifiers $\overline{U}$ returned by the unsorted $\overline{\mathcal{E}}$-unification procedure. The basic techniques behind our algorithm were described in [12]. However the correctness shown in [12] imposed conditions that are *too strong* when the theory $\mathcal{E}$ contains collapsing equations like identity or idempotence axioms.

Our approach to find suitable requirements is then to relax the requirements on $\mathcal{E}$ while making requirements on the unsorted unification procedure in relation to the theory $\mathcal{E}$. At first this appears to be less general than the approach in [12], since that work did not make any assumption about the unsorted unification procedure. However, as we will discuss later, the theories we are interested in are such that every practical unification procedure will satisfy the requirements. Most importantly for our work, this includes theories with identity axioms.

In this section, we assume the following conditions on the order-sorted theory $\mathcal{E} = (\Sigma, E)$ and the unsorted unification procedure for $\overline{\mathcal{E}}$.

(i) Σ is *preregular* [8], that is every term $t \in T_\Sigma(X)$ has a *least sort* $\mathrm{ls}(t) \in S$.

(ii) $\mathcal{E}$ is *sort-independent* which means that for all order-sorted terms $t, u \in T_\Sigma(X)$,

$$t =_{\overline{\mathcal{E}}} u \Rightarrow t =_{\mathcal{E}} u.$$

(iii) For each unification problem Γ, the unsorted unification procedure generates a complete finite set of most-general unifiers $\overline{U}$ which is *sort preserving*, which means that for each order-sorted unifier $\psi \in \mathrm{Un}_{\mathcal{E}}(\Gamma)$, there is an unsorted unifier $\overline{\theta} \in \overline{U}$ and unsorted substitution $\overline{\phi} : \mathrm{rvars}(\overline{\theta}) \to T_\Sigma(X)$ such that: (1) $\psi =_{\overline{\mathcal{E}}} \overline{\theta\phi}$, and (2) $\overline{\theta\phi}$ is an order-sorted substitution.

If the equational theory $\mathcal{E}$ and unsorted $\overline{\mathcal{E}}$-unification procedure satisfy the previous requirements, as we show below, the unsorted $\overline{\mathcal{E}}$-unification procedure can be used to solve order-sorted $\mathcal{E}$-unification problems. We can split the process of solving an order-sorted unification problem $\Gamma = t_1 =_{\mathcal{E}} u_1 \wedge \cdots \wedge t_n =_{\mathcal{E}} u_n$ into two phases: an *unsorted unification* phase and a *sort propagation* phase.

Unsorted Unification. First, we call the unsorted $\overline{\mathcal{E}}$-unification procedure on the unsorted $\overline{\mathcal{E}}$-unification problem $\overline{\Gamma} = t_1 =_{\overline{\mathcal{E}}} u_1 \wedge \cdots \wedge t_n =_{\overline{\mathcal{E}}} u_n$ to obtain a finite complete set of most-general sort-preserving unifiers $\overline{U}$ for $\overline{\Gamma}$.

Given an initial set of membership constraints D, we freely apply the rules below to obtain a final set of constraints D^*.

Intersection $$\{\, t : s_1 \wedge t : s_2 \wedge M \,\} \to \bigcup_{s \in \mathrm{glb}_\Sigma(s_1,s_2)} \{\, t : s \wedge M \,\}$$

Propagation $$\{\, f(t_1,\ldots,t_n) : s \wedge M \,\} \to \bigcup_{s_1 \ldots s_n \in \mathrm{ar}_\Sigma(f,s,n)} \{\, t_1 : s_1 \wedge \cdots \wedge t_n : s_n \wedge M \,\}$$

Subsumption $$\{\, M_1, M_2 \,\} \to \{\, M_1 \,\} \quad \text{if } M_1 \geq M_2$$

$$\begin{aligned} \text{where } \mathrm{glb}_\Sigma(s_1,s_2) &= \sup_{\leq}(\{\, s \in S \mid s \leq s_1 \wedge s \leq s_2 \,\}), \\ \mathrm{ar}_\Sigma(f,s,n) &= \sup_{\leq^n}(\{\, w \in S^n \mid (\exists s' \in S)\ f \in F_{w,s'} \wedge s' \leq s \,\}), \text{ and} \\ M_1 \geq M_2 &\iff (\forall t : s \in M_1)(\exists s' \in S)\ s' \leq s \wedge t : s' \in M_2. \end{aligned}$$

Fig. 1. Sort Propagation Algorithm

Sort Propagation. In the second phase, for each unsorted unifier $\overline{\theta} \in \overline{U}$, we use the *membership propagation* algorithm described below to generate a set of *variable renamings*. In this context, a variable renaming is an injective function $\rho : \mathrm{rvars}(\overline{\theta}) \to X$. For each variable renaming ρ generated for an unsorted unifier $\overline{\theta} \in \overline{U}$, our procedure returns $\overline{\theta}\rho$ as one element in the complete set of most-general unifiers.

The membership propagation algorithm is described by a set of rules that maintain a *disjunctive* set D of *membership constraints*. Each membership constraint $M \in D$ is a conjunctive formula of the form $M = t_1 : s_1 \wedge \cdots \wedge t_n : s_n$, and D is a finite set $D = \{\, M_1, \ldots, M_p \,\}$ of membership constraints. A membership constraint M captures constraints for an unsorted unifier to be an formed order-sorted unifier.

For each unsorted unifier $\overline{\theta} \in \overline{U}$, we initially generate a singleton set $D(\overline{\theta})$ reflecting the sort constraints on the variables appearing in the original unification problem Γ.

$$D(\overline{\theta}) = \{\, \bigwedge_{x_s \in \mathrm{vars}(\Gamma)} x_s\overline{\theta} : s \,\}.$$

We then apply the three rewrite rules in Fig. 1 to $D(\overline{\theta})$ until termination. The **Intersection** rule exploits the preregularity assumption to simplify multiple membership constraints $t : s_1$ and $t : s_2$ on the same term t. The **Propagation** rule simplifies constraints on terms $f(t_1,\ldots,t_n) : s$ to the smaller terms $t_1,\ldots,t_n$. Finally, the **Subsumption** rule is used to eliminate membership constraints that are subsumed by other more-general membership constraints. We let D^* denote the unique normal form obtained by rewriting D until completion.

Upon termination of the rules each membership constraint $M \in D^*$ will have the

form $x_1 : s_1 \wedge \cdots \wedge x_n : s_n$ where $x_i \neq x_j$ if $i \neq j$. We call membership constraints with this form *reduced*. A reduced membership constraint can be viewed as a function $\mathrm{sort}_M : \mathrm{rvars}(\overline{\theta}) \to S$ that maps each variable $x_i \in \mathrm{rvars}(\overline{\theta})$ to the sort $s_i \in S$. Furthermore, for each reduced membership constraint M, we let $\rho_M : \mathrm{rvars}(\overline{\theta}) \to X$ be a variable renaming which maps each variable $x \in \mathrm{rvars}(\overline{\theta})$ to a fresh variable $x\rho_M$ with sort $\mathrm{sort}_M(x)$.

For the set of unsorted sort-preserving unifiers $\overline{U} \subseteq \mathrm{Un}_{\overline{\mathcal{E}}}(\Gamma)$, we define the set

$$\mathrm{OS}(\overline{U}) = \{\, \overline{\theta}\rho_M \mid \overline{\theta} \in \overline{U} \wedge M \in D(\overline{\theta})^* \,\}.$$

As an example, consider the unification problem $x_{NzNat} = y_{Nat} + z_{Nat}$ over an order-sorted theory $\mathcal{E} = (F, E)$ where $+$ contains the following operator declarations:

$$+ : Nat\ Nat \to Nat \qquad + : Nat\ NzNat \to NzNat \qquad + : NzNat\ Nat \to NzNat$$

where the declaration $f : s_1\, s_2 \to s$ means that $f \in F_{s_1 s_2, s}$. In this case, the unsorted unification engine can return a single unifier

$$\overline{\theta} = (x \mapsto y + z, y \mapsto y, z \mapsto z).$$

However, $\overline{\theta}$ is not an order-sorted unifier, because $y + z$ does not have sort $NzNat$. We pass $\overline{\theta}$ to the sort propagation algorithm, which generates the initial set of membership constraints

$$D(\overline{\theta}) = \{\, y + z : NzNat \wedge y : Nat \wedge z : Nat \,\}.$$

For this simple example, a single application of **Propagation** yields the membership constraints:

$$\begin{aligned} D(\overline{\theta})' = \{\, &(y : Nat \wedge z : NzNat \wedge y : Nat \wedge z : Nat), \\ &(y : NzNat \wedge z : Nat \wedge y : Nat \wedge z : Nat) \,\}. \end{aligned}$$

From $D(\overline{\theta})'$, we only need to apply **Intersection** several times to yield the final set of membership constraints:

$$D(\overline{\theta})^* = \{\, (y : Nat \wedge z : NzNat), (y : NzNat \wedge z : Nat) \,\}.$$

From $D(\overline{\theta})^*$, we can extract two variables renamings. When applied to the initial unsorted unifier $\overline{\theta}$, this yields the final complete set of order-sorted unifiers:

$$\begin{aligned} \mathrm{OS}(\overline{U}) = \{\, &(x \mapsto u_{Nat} + v_{NzNat},\ y \mapsto u_{Nat},\ z \mapsto v_{NzNat}), \\ &(x \mapsto u_{NzNat} + v_{Nat},\ y \mapsto u_{NzNat},\ z \mapsto v_{Nat}) \,\}. \end{aligned}$$

We prove the following result in Section 5 to show that our algorithm is correct,

Theorem *Let $\mathcal{E} = (\Sigma, E)$ denote an order-sorted theory satisfying requirements (i) and (ii) above, then given a unification problem Γ with a complete set of most-*

general sort-preserving unsorted unifiers $\overline{U}$, $\mathrm{OS}(\overline{U})$ *is a complete set of most-general order-sorted unifiers for* Γ.

4 Order-sorted AC + ACU unification

As the requirements on $\mathcal{E}$ and $\overline{U}$ seem rather technical, to give the reader a more intuitive feel for them, we show how the requirements are satisfied by many order-sorted equational theories specified as Maude modules having free, commutative, AC, and ACU symbols. Essentially, each such Maude module can be viewed as an order-sorted theory $\mathcal{E} = (\Sigma, E)$ with $\Sigma = (S, F, \leq)$ such that:

(a) Each equivalence class $[s] \in S/\equiv_{\leq}$ contains a maximal element k_s called the *kind* of s where $\equiv_{\leq}$ denotes the equivalence relation generated by $\leq$. Moreover, for each operator declaration $f \in F_{s_1 \ldots s_n, s}$, there is also a declaration $f \in F_{k_{s_1} \ldots k_{s_n}, k_s}$.

(b) E contains axioms of the following forms:

$$f(f(x,y),z) = f(x,f(y,z)) \quad f(x,y) = f(y,x) \quad f(c,x) = x$$
$$\text{associativity} \qquad \text{commutativity} \qquad \text{unit}$$

where the sorts of x, y, z are maximal sorts, that is, sorts of the form k_s for some $s \in S$, and for each binary symbol $f \in F$, either f does not appear in E, or E contains commutativity (C), associativity and commutativity (AC), or associativity, commutativity, and unit (ACU) axioms for f.

(c) Σ is preregular.

(d) Each axiom $l = r \in E$ is *sort-preserving*, that is for each variable mapping $\rho : X \to X$, $\mathrm{ls}(l\rho) = \mathrm{ls}(r\rho)$.

The assumptions (a)–(d) are quite reasonable for order-sorted Maude specifications with free, commutative, AC, and ACU operators. Maude will automatically introduce additional top-most sorts k_s, and requires that associativity, commutativity, and unit axioms satisfy the requirements in (b). Maude does allow associative symbols that are not commutative, however unification for such theories may be infinitary [13] and is not considered here. The preregularity requirement is checked automatically by Maude when the module is entered. The sort-preservation requirement (d) is essential as the sort-propagation algorithm described in the previous section operates syntactically on terms, and disregards the possibility that applying an equation may change the sort of a term. It is guaranteed by a three-pronged approach:

- For each associativity axiom $f(f(x,y),z) = f(x,f(y,z))$, Maude checks that it is sort preserving by considering possible variable mappings.
- For each commutativity axiom $f(x,y) = f(y,x)$ and each declaration $f : s_1 s_2 \to s$, Maude completes the theory by adding a declaration $f : s_2\, s_1 \to s$.
- For each pair of identity axioms $f(x,c) = x$ and $f(c,x) = x$, our unification

procedure completes the theory by introducing a fresh sort s_c together with: (1) an operator declaration $c : \to s_c$, (2) a subsort declaration $s_c < \mathrm{ls}(t)$; and (3) for each sort $s \in [\mathrm{ls}(t)]$, operator declarations $f : s\ s_c \to s$ and $f : s_c\ s \to s$.

We now focus on the relationship between the assumptions (a)–(d) and the earlier requirements (i)–(iii). The first preregularity requirement follows from the preregularity assumption. The sort-independence requirements follows form the assumptions (a) and (b).

Theorem 4.1 *If $\mathcal{E} = (\Sigma, E)$ is an order-sorted theory satisfying assumptions (a) and (b) above, then $\mathcal{E}$ is sort-independent.*

Proof. Showing that $\mathcal{E}$ is sort-independent requires showing that for all $t, u \in T_\Sigma(X)$, $t =_{\overline{\mathcal{E}}} u$ implies $t =_{\mathcal{E}} u$.

We first partition E into disjoint sets $E = R \uplus A_x$ where A_x contains the associativity and commutativity equations in E and the identity equations $f(c, x) = x$ in E are interpreted as rules $f(c, x) \to x$ in R. It is not difficult to see that the rules R modulo A_x are terminating and confluent, and therefore $t =_{\overline{\mathcal{E}}} u$ iff $t{\downarrow_{\overline{R}/\overline{A}_x}} =_{\overline{A}_x} u{\downarrow_{\overline{R}/\overline{A}_x}}$.

As A_x only contains associativity and commutativity axioms, if $t \in T_\Sigma(X)_{k_s}$ for some maximal sort k_s and $t =_{\overline{A}_x} v$, then it easily follows that $v \in T_\Sigma(X)_{k_s}$ and $t =_{A_x} v$ by the requirement (a). It also easily follows that if $t \to^*_{\overline{R}/\overline{A}_x} v$, then $v \in T_\Sigma(X)_{k_s}$ and $t \to^*_{R/A_x} v$. From this, we can conclude that

$$t{\downarrow_{\overline{R}/\overline{A}_x}} =_{\overline{A}_x} u{\downarrow_{\overline{R}/\overline{A}_x}} \implies t{\downarrow_{R/A_x}} =_{A_x} u{\downarrow_{R/A_x}}.$$

It easily follows that $t =_{\mathcal{E}} u$, and thus $\mathcal{E}$ is sort-independent. □

In general, the requirement that the unification procedure is sort-preserving *does not* follow from the assumptions given above. For an example, consider the theory $\mathcal{E}$ with two unrelated top-most sorts *Nat* and *Cns* where *Nat* contains the ACU symbol + with the identity element 0, and *Cns* contains the constant a. Given the unification problem $x_{Cns} = a$, it would be permissible for the unsorted unification procedure to return the unifiers

$$\overline{U} = \{\, (x \mapsto a + 0) \,\}.$$

This is a complete set of unsorted unifiers due to the identity axiom, but unsuitable for our sort propagation algorithm as $a + 0$ is not a legal term. This counterexample illustrates why the earlier work [12] imposed significant restrictions on theories with collapsing equations like identity.

These stronger restrictions appear unnecessary in practice — in our experience, the procedure will not introduce extra symbols, and in this case return the simpler unifier $x \mapsto a$. The reason that unsorted AC and ACU unification procedures satisfy this assumption is that the unifiers are computed from the terms appearing in equations $l = r \in \Gamma$. When those subterms are well-typed with the same top-most sort k, substitutions generated by the unsorted unification procedure should

be well-typed as well Provided that the sorts of fresh variables in the right hand side of a variable are given the appropriate top sort k, due to our assumption (a), we have found it is safe to assume the following:

(e) For each unifier $\overline{\theta}$ in the set of unifiers $\overline{U}$ returned by the unsorted unification procedure for the order-sorted unification problem Γ, and for each variable $x_s \in \text{vars}(\Gamma)$, $x_s\overline{\theta} \in T_\Sigma(X)_{k_s}$.

To validate these ideas and test this assumption, we have extended an alpha version of Maude so that it may communicate with CiME [3,4] by passing unsorted unification problems as strings, and parsing the unsorted unifiers returned from CiME back into Maude terms. As an additional safeguard, the parsing process checks the substitutions returned by CiME to verify that assumption (e) is satisfied. These checks have always been satisfied in our experience using the procedure so far. We then apply the sort propagation algorithm described in the previous section to generate order-sorted $\mathcal{E}$-unifiers. The order-sorted unification procedure is used to analyze cryptographic protocols with algebraic properties of associativity and commutativity using the Maude-NRL protocol analyzer [5].

5 Correctness Proof

The goal of this section is to show the correctness of our approach to order-sorted equational unification. Before we can show this, we need several intermediate lemmas. The first lemma shows how preregularity is used.

Lemma 5.1 *If $\Sigma = (S, F, \leq)$ is preregular, then for all sorts $s_1, s_2 \in S$ and terms $t \in T_{\overline{\Sigma}}(\overline{X})$,*

$$t \in T_\Sigma(X)_{s_1} \cap T_\Sigma(X)_{s_2} \iff (\exists s \in \text{glb}_\Sigma(s_1, s_2))\, t \in T_\Sigma(X)_s$$

where $\text{glb}_\Sigma(s_1, s_2) = \sup_{\leq}(\{\, s \in S \mid s \leq s_1 \wedge s \leq s_2 \,\})$.

Proof. If there is a sort $s \in \text{glb}(s_1, s_2)$ such that $t \in T_\Sigma(X)_s$, then $t \in T_\Sigma(X)_{s_1} \cap T_\Sigma(X)_{s_2}$ as $s \leq s_1$ and $s \leq s_2$. We still must show that $t \in T_\Sigma(X)_{s_1} \cap T_\Sigma(X)_{s_2}$ implies that there is a sort $s \in \text{glb}(s_1, s_2)$ such that $t \in T_\Sigma(X)_s$. However, this follows immediately as t must have a least sort $s' \in S$. It must be the case that $s' \leq s_1$ and $s' \leq s_2$. Therefore, there is an $s \in \text{glb}(s_1, s_2)$ such that $s' \leq s$. As $T_\Sigma(X)_{s'} \subseteq T_\Sigma(X)_s$, it follows that $t \in T_\Sigma(X)_s$. □

Lemma 5.2 *For all terms $f(t_1, \ldots, t_n) \in T_{\overline{\Sigma}}(\mathrm{X})$ and sorts $s \in S$,*

$$f(t_1, \ldots, t_n) \in T_\Sigma(X)_s \iff (\exists s_1 \ldots s_n \in \text{ar}_\Sigma(f, s, n))\, t_1 \in T_\Sigma(X)_{s_1} \wedge \cdots \wedge t_n \in T_\Sigma(X)_{s_n}$$

where $\text{ar}_\Sigma(f, s, n) = \sup_{\leq^n}(\{\, w \in S^n \mid (\exists s' \in S)\, f \in F_{w,s'} \wedge s' \leq s \,\})$.

Proof. If there are sorts $s_1 \ldots s_n \in \text{ar}_\Sigma(f, s, n)$ such that $t_i \in T_\Sigma(X)_{s_i}$ for $i \in [1, n]$, then there must be a sort $s' \leq s$ such that $f \in F_{s_1 \ldots s_n, s'}$. It follows that $f(t_1, \ldots, t_n) \in T_\Sigma(X)_{s'}$, and thus $f(t_1, \ldots, t_n) \in T_\Sigma(X)_s$.

On the other hand, if $f(t_1, \ldots, t_n) \in T_\Sigma(X)_s$, then there is some $s' \leq s$ such that $f \in F_{s'_1 \ldots s'_n, s'}$ and $t_i \in T_\Sigma(X)_{s'_i}$ for $i \in [1, n]$. It follows that there are sorts $s_1 \ldots s_n \in \mathrm{ar}_\Sigma(f, s, n)$ such that $s'_i \leq s_i$ for $i \in [1, n]$. Consequently, $t_i \in T_\Sigma(X)_{s_i}$ for $i \in [1, n]$. □

For a membership constraint M, we define the *unifiers* for M, denoted $\mathrm{Un}_{\overline{\Sigma}}(M)$ to be the set of unsorted substitutions $\theta : \overline{X} \to T_{\overline{\Sigma}}(\overline{X})$ such that for each membership $t : s \in M$, $t\theta \in T_\Sigma(X)_s$.

Lemma 5.3 *For each order-sorted signature $\Sigma = (S, F, \leq)$ and pair of membership constraints M_1 and M_2,*

$$M_1 \geq M_2 \implies \mathrm{Un}_\Sigma(M_1) \supseteq \mathrm{Un}_\Sigma(M_2).$$

Proof. To show that $\mathrm{Un}_\Sigma(M_1) \supseteq \mathrm{Un}_\Sigma(M_2)$, we must show for each substitution $\theta \in \mathrm{Un}_\Sigma(M_2)$ and membership $t : s \in M_1$, we have $t\theta \in T_\Sigma(X)_s$. However, since $M_1 \geq M_2$, we know that for each $t : s \in M_1$, there is a membership $t : s' \in M_2$ such that $s' \leq s$. By definition $t\theta \in T_\Sigma(X)_{s'}$, and therefore $t\theta \in T_\Sigma(X)_s$. □

When the membership constraints M_1 and M_2 are reduced, the previous implication holds in the other direction.

Lemma 5.4 *For each order-sorted signature $\Sigma = (S, F, \leq)$ and pair of reduced membership constraints M_1 and M_2 such that* $\mathrm{vars}(M_1) = \mathrm{vars}(M_2)$*,*

$$\mathrm{Un}_\Sigma(M_1) \supseteq \mathrm{Un}_\Sigma(M_2) \implies M_1 \geq M_2$$

Proof. Since both M_1 and M_2 are reduced and $\mathrm{vars}(M_1) = \mathrm{vars}(M_2)$, to show that $M_1 \geq M_2$, it is sufficient to show that for each $x \in \mathrm{vars}(M_1)$, $\mathrm{sort}_{M_1}(x) \geq \mathrm{sort}_{M_2}(x)$. Since M_2 is reduced, there is a substitution $\rho_{M_2} : \mathrm{vars}(M_2) \to T_\Sigma(X)$ which maps each variable $x \in \mathrm{vars}(M_2)$ to the a fresh variable x' with sort $\mathrm{sort}_{M_2}(x) \in S$. Clearly $\rho_{M_2} \in \mathrm{Un}_\Sigma(M_2)$, and so $\rho_{M_2} \in \mathrm{Un}_\Sigma(M_1)$ by assumption. It follows that for each $x \in \mathrm{vars}(M_2)$ that $\mathrm{sort}_{M_1}(x) \geq \mathrm{sort}_{M_2}(x)$ since $x\rho_{M_2}$ is a variable with sort $\mathrm{sort}_{M_2}(x)$ and $x\rho_{M_2} \in \mathrm{Un}_\Sigma(M_2)$. □

For a disjunctive set of membership constraints D, we let $\mathrm{Un}_{\overline{\Sigma}}(D)$ denote the set of unsorted substitutions that are unifiers for a set of membership constraints $M \in D$, i.e.,

$$\mathrm{Un}_{\overline{\Sigma}}(D) = \bigcup_{M \in D} \mathrm{Un}_{\overline{\Sigma}}(M).$$

The key correctness property of the inference rules in Fig. 1 is captured by the following lemma.

Lemma 5.5 *For an preregular order-sorted signature Σ, if $D_1 \to^* D_2$ using the inference rules in Fig. 1, then* $\mathrm{Un}_\Sigma(D_1) = \mathrm{Un}_\Sigma(D_2)$.

Proof. To show this it is enough to show the single step case that $D_1 \to D_2$ implies $\mathrm{Un}_\Sigma(D_1) = \mathrm{Un}_\Sigma(D_2)$. The full lemma follows easily by induction on the number of

rules used to show $D_1 \to^* D_2$. To show the single step case, we must consider three separate cases, one for each of the inference rules in Fig. 1.

- If **Intersection** is used, we know that D_1 and D_2 have the forms $D_1 = \{ t : s_1 \wedge t : s_2 \wedge M \} \cup D$ and $D_2 = \bigcup_{s \in \mathrm{glb}_\Sigma(s_1,s_2)} \{ t : s \wedge M \} \cup D$. However, for each order-sorted substitution θ we know by Lemma 5.1 that $t\theta \in T_\Sigma(X)_{s_1} \cap T_\Sigma(X)_{s_2}$ iff there is a sort $s \in \mathrm{glb}_\Sigma(s_1, s_2)$ such that $t\theta \in T_\Sigma(X)_s$. It follows that $\mathrm{Un}(D_1) = \mathrm{Un}(D_2)$.
- If **Propagation** is used, we know that D_1 and D_2 have the forms $D_1 = \{ f(t_1, \ldots, t_n) : s \wedge M \} \cup D$ and $D_2 = \bigcup_{s_1 \ldots s_n \in \mathrm{ar}_\Sigma(f,s)} \{ t_1 : s_1 \wedge \cdots \wedge t_n : s_n \wedge M \} \cup D$. However, for each order-sorted substitution θ we know by Lemma 5.2 that $f(t_1\theta, \ldots, t_n\theta) \in T_\Sigma(X)_s$ iff there are sorts $(s_1 \ldots s_n) \in \mathrm{ar}_\Sigma(f, s)$ such that $t_i\theta \in T_\Sigma(X)_{s_i}$ for $i \in [1, n]$.
- If **Subsumption** is used, we know that D_1 and D_2 have the forms $D_1 = \{ M_1, M_2 \} \cup D$ and $D_2 = \{ M_1 \} \cup D$ where $M_1 \geq M_2$. However, it follows easily that $\mathrm{Un}(D_1) = \mathrm{Un}(D_2)$ as for each substitution $\mathrm{Un}(M_2) \subseteq \mathrm{Un}(M_1)$ by Lemma 5.3.

□

In order to preserve the set of substitutions, we also need to show that the inference rules do not discard variables or introduce new ones:

Lemma 5.6 *If $D_1 \to^* D_2$ using the inference rules in Fig. 1 and each set of membership constraints $M_1 \in D_1$ has the same variables $\mathrm{vars}(M_1) = X$, then for all $M_2 \in D_2$, $\mathrm{vars}(M_2) = X$.*

Proof. This is a straightforward induction over the number of rewrites used to show $D_1 \to^* D_2$ and considering each rule separately. □

The following lemma is useful to show that the inference rules terminate with a unique set of membership constraints.

Lemma 5.7 *If D_1 and D_2 are both disjunctive sets of membership constraints that are irreducible by the inference rules in Fig. 1, then $\mathrm{Un}_\Sigma(D_1) = \mathrm{Un}_\Sigma(D_2)$ implies $D_1 = D_2$.*

Proof. We show this by showing that $D_1 \neq D_2$ implies $\mathrm{Un}(D_1) \neq \mathrm{Un}(D_2)$. If $D_1 \neq D_2$, then there must be a conjunction of membership constraints M_1 which is in $D_1 \setminus D_2$ or $D_2 \setminus D_1$. We assume that the M_1 is in D_1 as the other case is symmetric. Since the rules in Fig. 1 cannot be applied to D_1, we know that M_1 must be reduced, and hence has the form $M_1 = x_1 : s_1 \wedge \cdots \wedge x_n : s_n$ with $x_i \neq x_j$ for each $i \neq j$. Let ρ_{M_1} denote the substitution mapping each variable $x \in \mathrm{vars}(M_1)$ to a fresh variable $x\rho_{M_1}$ with sort $\mathrm{sort}_{M_1}(x) \in S$. By definition $\rho_{M_1} \in \mathrm{Un}_\Sigma(M_1)$ and therefore $\rho_{M_1} \in \mathrm{Un}(D_1)$. If $\rho_{M_1} \notin \mathrm{Un}(D_2)$, then $\mathrm{Un}(D_1) \neq \mathrm{Un}(D_2)$, and consequently we are done. Otherwise, ρ_{M_1} is in $\mathrm{Un}(D_2)$, and so there must be a membership $M_2 \in \mathrm{Un}(D_2)$ such that for each membership $x : s \in M_2$ there is a membership $x : s' \in M_1$ with $s' \leq s$. It follows that $M_2 \geq M_1$. Since D_1 is fully reduced by the rules in Fig. 1, it follows that the substitution ρ_{M_2} is not in

$\mathrm{Un}_\Sigma(D_1)$, since this would imply that there is a mapping $M \geq M_2 \geq M_1$ in D_1. This is impossible since D_1 has been fully reduced by the rules in Fig. 1. □

Using the previous lemmas, it is not difficult to show the following Termination Theorem which shows that the inference rules terminate with a unique set of membership constraints.

Theorem 5.8 (Termination Theorem) *For each disjunctive set of membership constraints D, there is a unique set of membership constraints D^* such that $D \to^! D^*$ using the inference rules in Fig. 1.*

Proof. Showing this requires proving that: (1) the rules in Fig. 1 are terminating and (2) if $D \to^! D_1$ and $D \to^! D_2$, then $D_1 = D_2$. The rules in Fig. 1 are terminating, because each rewrite either reduces the size of a term in a membership, or preserves the terms while reducing the total number of memberships. To show (2), observe that if $D \to^! D_1$ and $D \to^! D_2$, then $\mathrm{Un}_\Sigma(D_1) = \mathrm{Un}_\Sigma(D) = \mathrm{Un}_\Sigma(D_2)$ by Lemma 5.5. Therefore, $D_1 = D_2$ by Lemma 5.7. □

We are now ready to conclude with a proof of the main theorem of the paper:

Theorem 5.9 *Let $\mathcal{E} = (\Sigma, E)$ denote a preregular and sort-independent order-sorted theory, then given a unification problem Γ with a complete set of most-general sort-preserving unsorted unifiers $\overline{U}$, $\mathrm{OS}(\overline{U})$ is a complete set of most-general order-sorted unifiers for Γ.*

Proof. Since $\mathcal{E}$ is sort-independent, we can assume that E can be partitioned into rewrite rules R and equations A such that $\overline{R}$ is confluent and terminating modulo $\overline{A}$. Moreover, we can assume that each substitution $\overline{\theta} \in \overline{U}$ is $\overline{R}/\overline{A}$-irreducible.

Proving the above theorem requires showing three things: (1) each element of $\mathrm{OS}(\overline{U})$ is an order-sorted unifier for Γ; (2) the set of unifiers $\mathrm{OS}(\overline{U})$ is complete; (3) the set of unifiers $\mathrm{OS}(\overline{U})$ is most-general. We show each of these facts separately.

- For each element $\theta \in \mathrm{OS}(\overline{U})$, there is an unsorted unifier $\overline{\theta} \in \overline{U}$ and reduced membership constraints $M \in D(\overline{\theta})^*$ such that $\theta = \overline{\theta}\rho_M$. We first show that $x_s\theta \in T_\Sigma(X)_s$ for each variable $x_s \in \mathrm{vars}(\Gamma)$. To see this, observe that by definition $\rho_M \in \mathrm{Un}_\Sigma(D(\overline{\theta})^*)$, and so by Lemma 5.5, $\rho_M \in \mathrm{Un}_\Sigma(D(\overline{\theta}))$. Furthermore, by Lemma 5.6, we know that $\mathrm{vars}(M) = \mathrm{vars}(D(\overline{\theta})) = \mathrm{vars}(\Gamma)$. It follows by definition that for each variable $x_s \in \mathrm{vars}(\Gamma)$, $x\theta = x\overline{\theta}\rho_M$ is in $T_\Sigma(X)_s$. For each equation $t = u$ in Γ, we know that both t and u are well-sorted terms belonging to the same component. It follows that $t\theta$ and $u\theta$ are well-sorted terms with the same connected component. By definition $t\theta =_{\overline{\mathcal{E}}} u\theta$, and as $\mathcal{E}$ is sort-independent it follows that $t\theta =_{\mathcal{E}} u\theta$.
- To show that $\mathrm{OS}(\overline{U})$ is complete, we must show for each order-sorted unifier $\psi \in \mathrm{Un}_{\mathcal{E}}(\Gamma)$, there is a unifier $\theta \in \mathrm{OS}(\overline{U})$ and order-sorted substitution $\phi : \mathrm{rvars}(\theta) \to T_\Sigma(X)$ such that $x\psi =_{\mathcal{E}} x\theta\phi$ for each $x \in \mathrm{vars}(\Gamma)$. Let ψ be a unifier in $\mathrm{Un}_{\mathcal{E}}(\Gamma)$. As $\overline{U}$ is a complete set of sort-preserving unifiers, there is an unifier $\overline{\theta} \in \overline{U}$ such that $\psi =_{\overline{\mathcal{E}}} \overline{\theta}\overline{\phi}$ for some unsorted substitution $\overline{\phi} : Y \to T_\Sigma(X)$ with

$Y = \text{rvars}(\overline{\theta})$. Moreover, since $x_s\psi \in T_\Sigma(X)_s$ for each variable $x_s \in \text{vars}(\Gamma)$, we can assume that $x_s\overline{\theta\phi} \in T_\Sigma(X)_s$ since $\overline{U}$ is sort-preserving.

It follows that $\overline{\phi} \in \text{Un}_\Sigma(D(\overline{\theta}))$. By Lemma 5.5 and Theorem 5.8, there must be a reduced set of membership constraints $M \in D(\overline{\theta})^*$ such that

$$(\forall x \in Y)\; x\overline{\phi} \in T_\Sigma(X)_{\text{sort}_M(x)}. \tag{1}$$

Since M is reduced, there is a variable renaming ρ_M with maps each variable $x \in Y$ to a fresh variable x' with sort $\text{sort}_M(x)$. Let ρ_M^{-1} denote the inverse of that renaming. By using (1), it should be clear that $\rho_M^{-1}; \overline{\phi}$ is an order-sorted substitution. Moreover, as $\rho_M \in \text{Un}_\Sigma(M)$ and therefore in $\text{Un}_\Sigma(D(\overline{\theta}))$ by Lemma 5.5, $\overline{\theta}; \rho_M$ must be an order-sorted substitution. Since $\overline{\theta}; \rho_M \in \text{OS}(\overline{U})$ and $\psi = (\overline{\theta}; \rho_M); (\rho_M^{-1}; \overline{\phi})$, it follows that $\text{OS}(\overline{U})$ is a complete set of unifiers.

- To show that $\text{OS}(\overline{U})$ is a most-general set of unifiers, we must show for all distinct substitutions $\theta_1, \theta_2 \in \text{OS}(\overline{U})$, we have $\theta_1 \not\geq \theta_2$. We prove this by contradiction. Assume there are substitutions $\theta_1, \theta_2 \in \text{OS}(\overline{U})$ and a substitution $\psi : Y \to T_\Sigma(X)$ such that $\theta_1 = \theta_2; \psi$, where Y denotes the variables in the right-hand side of θ_2. Since both θ_1 and θ_2 are in $\text{OS}(\overline{U})$, we know they must have the form $\theta_1 = \overline{\theta_1}; \rho_{M_1}$ and $\theta_2 = \overline{\theta_2}; \rho_{M_2}$ with $\overline{\theta_1}, \overline{\theta_2} \in \overline{U}$, $M_1 \in D(\overline{\theta_1})^*$, and $M_2 \in D(\overline{\theta_2})^*$. Our assumption $\theta_1 = \theta_2; \psi$ implies that $\overline{\theta_1} = \overline{\theta_2}; (\rho_{M_2}; \psi; \rho_{M_1}^{-1})$. Since $\overline{U}$ is most-general, this can only be the case if $\overline{\theta_1} = \overline{\theta_2}$. As we assumed that $\theta_1 = \theta_2; \psi$, it is not difficult to show that $\psi = \rho_{M_2}^{-1}\rho_{M_1}$. For each variable $x \in Y$, we know that the sort of the variable $x\rho_{M_2}^{-1}\rho_{M_1}$ is $\text{sort}_{M_1}(x\rho_{M_2}^{-1})$, and $\text{sort}_{M_1}(x\rho_{M_2}^{-1}) \leq \text{sort}_{M_2}(x\rho_{M_2}^{-1})$ since ψ is an order-sorted substitution. It follows that $M_2 \geq M_1$ which is impossible since both $M_1, M_2 \in D(\overline{\theta_1})^*$ and $D(\overline{\theta_1})^*$ have been fully normalized using the inference rules in Fig. 1.

□

6 Conclusions and Related Work

There is a considerable amount of research already in unification in theories with sorts and subsorts (e.g., [1,7,9,12,14,15]) due to the improved expressiveness of order-sorted algebras and ability to simplify automated reasoning. The use of rule-based algorithms in describing unification has a long history as well with the most well-known example being Martelli and Montanari's algorithm for syntactic unification [10]. Our use of a rule-based approach to order-sorted unification is not particularly novel; however we wanted to revisit order-sorted equational unification after discovering that the soundness for the AC and ACI unification problems we were trying to solve did not follow from previous results.

Since our initial implementation of order-sorted unification in the Maude-NRL analyzer, an order-sorted unification engine has been included in the most recent Maude release [2]. This unification engine makes similar assumptions about the supported theories, however it has better performance as it will no longer need to parse unsorted unifiers back as strings, and can more tightly integrate the order-

sorted constraints into the core unification routines. This engine uses BDDs to symbolically solve the sort constraints, and has the advantage that the subsumption checks can be handled automatically by the BDD generation-algorithms.

Our aim in this paper is more general than the Maude-based applications of our algorithm. Our aim is one of *modularity*, so that different formal tool building efforts needing equational order-sorted unification procedures may be able to modularly decompose such a procedure into its *unsorted part* where several existing tools may be used and the rule-based *sort propagation algorithm* that we have presented and proved correct.

Acknowledgement

The authors would like to thank the referees for comments which helped to improve the paper.

References

[1] Boudet, A., *Unification in order-sorted algebras with overloading*, in: D. Kapur, editor, *Proc. of CADE 1992*, Lecture Notes in Comp. Sci. **607** (1992), pp. 193–207.

[2] Clavel, M., F. Durán, S. Eker, S. Escobar, P. Lincoln, N. Martí-Oliet, J. Meseguer and C. L. Talcott, *Unification and narrowing in Maude 2.4*, in: R. Treinen, editor, *RTA*, Lecture Notes in Computer Science **5595** (2009), pp. 380–390.

[3] Contejean, E. and C. Marché, *CiME: Completion modulo E*, in: H. Ganzinger, editor, *Proc. of RTA'96*, Lecture Notes in Comp. Sci. **1103** (1996), pp. 416–419.

[4] Contejean, E., C. Marché, B. Monate and X. Urbain, *CiME 2*, Available at: http://cime.lri.fr.

[5] Escobar, S., J. Hendrix, C. Meadows and J. Meseguer, *Diffie-Hellman cryptographic reasoning in the Maude-NRL protocol analyzer*, in: *Proc. of SecRet 2007*, 2007.

[6] Escobar, S., C. Meadows and J. Meseguer, *Equational cryptographic reasoning in the Maude-NRL protocol analyzer*, Electr. Notes Theor. Comput. Sci. **171** (2007), pp. 23–36.

[7] Gallier, J. and T. Isakowitz, *Order-sorted rigid E-unification*, Technical Report STERN IS-91-40, Stern School of Business at New York University (1991), available at http://hdl.handle.net/2451/14397.

[8] Goguen, J. A. and J. Meseguer, *Order-sorted algebra i: Equational deduction for multiple inheritance, overloading, exceptions and partial operations*, Theor. Comput. Sci. **105** (1992), pp. 217–273.

[9] Kirchner, C., *Order-sorted equational unification*, Presented at the fifth International Conference on Logic Programming (Seattle, USA) (1988), also as rapport de recherche INRIA 954, Dec. 88.

[10] Martelli, A. and U. Montanari, *An efficient unification algorithm*, ACM Trans. Program. Lang. Syst. **4** (1982), pp. 258–282.

[11] Meseguer, J., *Membership algebra as a logical framework for equational specification*, in: F. Parisi-Presicce, editor, *Proc. of WADT'97*, Lecture Notes in Comp. Sci. **1376** (1997), pp. 18–61.

[12] Meseguer, J., J. A. Goguen and G. Smolka, *Order-sorted unification*, J. Symbolic Computation **8** (1989), pp. 383–413.

[13] Plotkin, G., *Building-in equational theories*, Machine Intelligence **7** (1972), pp. 73–90.

[14] Walther, C., *Many-sorted unification*, J. ACM **35** (1988), pp. 1–17.

[15] Weidenbach, C., *Unification in sort theories and its applications*, Annals of Mathematics and Artificial Intelligence **18** (1996), pp. 261–293.