Order-Sorted Unification

JOSE MESEGUER, JOSEPH A. GOGUEN

SRI International, Menlo Park CA 94025

Center for the Study of Language and Information, Stanford University 94305

GERT SMOLKA

Universität Kaiserslautern, 6750 Kaiserslautern, West Germany

(Received 22 February 1988)

This paper studies unification for order-sorted equational logic. This logic generalizes unsorted equational logic by allowing a partially ordered set of sorts, with the ordering interpreted as set-theoretic containment in the models; it also allows overloading of function symbols, such as $+$ for integer and rational number addition, with the overloaded functions of greater rank interpreted in the models as extensions of those of smaller rank. Our presentation emphasizes semantic aspects, and gives a categorical treatment of unification that has substantial advantages in this context over the usual treatment of unifiers as endomorphisms of a single free algebra. Given system $\Gamma$ of equations and a set $E$ of axioms that is sort-preserving and does not impose restrictions on the sorts of its variables, the main results characterize when an order-sorted signature has a minimal (or finite, or most general when $\Gamma$ is solvable) family of order-sorted $E$-unifiers for $\Gamma$. In addition, for unitary signatures, where each solvable system of equations has a most general unifier, we give a quasi-linear algorithm for syntactic unification (i.e., for $E=\emptyset$) a la Martelli-Montanari, that is more efficient than the unsorted one for failures.

1 Introduction

Unification is interesting for a wide variety of applications in a wide variety of guises. The applications include "unification grammars" in linguistics (see [Shieber 86]), algorithms like resolution [Robinson 65] and Knuth-Bendix completion [Knuth & Bendix 70] for theorem proving (including so-called inductionless induction [Musser 80, Goguen 80, Huet & Hullot 82]), and programming languages such as Prolog [Colmerauer, Kanoui & van Caneghem 79] and Eqlog [Goguen & Meseguer 86] among many others. The guises include classical unification, which goes back to Herbrand [Herbrand 30], unification modulo equations, which...
goes back to Plotkin [Plotkin 72], and unification with subsorts, here called order-sorted unification, which has been studied by Schmidt-Schauss [Schmidt-Schauss 86] and Walther [Walther 85] [Walther 88] among others; Siekmann [Siekmann 88] gives a comprehensive survey of unification; see also [Fages & Huet 86]. Two other interesting guises are the CIL-unification of Mukai [Mukai 85] and the $\psi$-unification of Ait-Kaci [Ait-Kaci 84, A'it-Kaci 86], both of which are oriented towards applications to knowledge representation and natural language, and in the case of CIL-unification, particularly towards situation semantics [Barwise & Perry 83]. [Smolka & Ait-Kaci 87] shows how $\psi$-unification can be based on order-sorted logic and develops a framework in which order-sorted unification and $\psi$-unification coexist. An application of order-sorted unification that seems to have escaped prior notice is to polymorphism for typed functional languages (in the sense of Milner [Harper, MacQueen & Milner 86] and as implemented in ML), in the case where there are subtypes (i.e., subsorts) as well as polymorphic type constructors; for more on this matter see [Goguen 88], 7.5, and the polymorphic order-sorted logic approach given by Smolka [Smolka 88].

We feel that order-sorted unification has great potential for enhancing both efficiency and ease of use in many application areas, including theorem proving, knowledge representation, and linguistics. In particular, some future work will show how order-sorted unification can encompass both $\psi$-unification and CIL-unification [Goguen & Meseguer 88a].

This paper develops the theory of order-sorted unification on the semantic basis of order-sorted algebra in the form developed in [Goguen & Meseguer 87a, Goguen & Meseguer 88b]. Given a system $\Gamma$ of equations and a set $E$ of axioms that is sort-preserving and does not impose restrictions on the sorts of its variables, the main results characterize when an order-sorted signature has a minimal (or finite, or most general when $\Gamma$ is solvable) family of order-sorted $E$-unifiers for $\Gamma$. This greatly generalizes a characterization by Walther [Walther 86] for order-sorted signatures with $E=\emptyset$ and without overloaded function symbols, since we consider unification modulo axioms $E$ over arbitrary order-sorted signatures with (possible) overloading.

The usual treatment of unifiers as endomorphisms of a free algebra on many generators is syntactically complex, and in the order-sorted case can even be misleading. For example, the characterization theorem of [Walther 86], that most general unifiers exist iff the sorts actually form a forest, is misleading because of using an insufficiently general notion of most general unifier, arrived at by viewing unifiers as endomorphisms of a single free algebra. This motivates our categorical treatment of unification, which is simpler, clearer, applicable to a wide variety of unification frameworks besides the order-sorted one, and neatly captures uniqueness properties of unification through initiality or related properties.

For unitary signatures, where each solvable system of equations has a most general unifier, we give a quasi-linear algorithm for syntactic unification (i.e., for $E=\emptyset$) a la Martelli and Montanari [Martelli & Montanari 82], that is more efficient than the unsorted one for failures.
Efficient failure detection can have an enormous impact in practical applications such as logic programming.

The relation between order-sorted and unsorted unifiers was originally considered by Schmidt-Schauss [Schmidt-Schauss 86] in a syntactically oriented framework. Our approach, in contrast, uses a framework firmly based on model-theoretic semantics, including a completeness theorem for order-sorted logic [Goguen & Meseguer 87b]. We extend Schmidt-Schauss' [Schmidt-Schauss 86] results in the following respects:

- We give a decidable sufficient condition for when order-sorted unification can be related to unsorted unification. (Schmidt-Schauss gives an undecidable necessary and sufficient condition.)
- We give new characterization theorems for when there exist families of E-unifiers that are unique, that are finite, and that are minimal, at both the unsorted and many-sorted levels.
- We consider potentially infinite signatures, which are important for the characterization theorems, whereas Schmidt-Schauss only considers finite signatures.

The present paper is a slightly improved version of Report CSLI-87-86 (Center for the Study of Language and Information, Stanford University, March 1987) bearing the same title, that was distributed at the Val d’Aojol workshop on unification in March 1987, and at the Lakeway, Texas, Colloquium on the Resolution of Equations in Algebraic Structures in May 1987.

2 Order-Sorted Algebra

We assume familiarity with S-sorted sets and functions (for S a set of sorts) and with many-sorted algebra (which may be abbreviated MSA hereafter) signatures (S,Σ), with Σ-algebras and with Σ-homomorphisms (see for example [Meseguer & Goguen 85]). We now proceed to generalize these concepts to order-sorted algebra (which may be abbreviated OSA hereafter); further details of order-sorted algebra can be found in [Goguen & Meseguer 87a], and [Goguen & Meseguer 88b] which also contains a discussion of other approaches. A very complete exposition of a useful alternative approach with a somewhat different semantics can be found in [Smolka, Nutt, Goguen & Meseguer 88]. The subject of order-sorted algebra was initiated by the paper [Goguen 78].

Definition 1: An order-sorted signature is a triple (S,≤,Σ) such that (S,Σ) is an MSA signature, (S,≤) is a partially ordered set, and the operators satisfy the following monotonicity condition:

\[
\text{if } \sigma \in \Sigma_{w_1,s_1} \land \Sigma_{w_2,s_2} \text{ and if } w_1 \leq w_2, \text{ then } s_1 \leq s_2.
\]

We extend the ordering ≤ on S to strings of equal length in S* by \(s_1 \leq s_2\) if \(s_1s_2' \leq s_2s_2'\) for \(1 \leq i \leq n\). Similarly, ≤ extends to pairs \((w,s)\) in S*×S by \((w,s) \leq (w',s')\) iff \(ws \leq w's'\) and \(ss'\).

Although not needed for validity of our results, this very natural condition rules out some bizarre models.
When the sort set $S$ is clear, we write $\Sigma$ for $(S,\Sigma)$. Similarly, when the partial order $(S,\leq)$ is clear, we write $\Sigma$ for $(S,\leq,\Sigma)$. Also, we may write $\sigma: w \rightarrow s$ for $\sigma \in \Sigma_{w,s}$ to emphasize that $\sigma$ denotes a function with arity $w$ and value sort (or co-arity) $s$. An important special case is $w=\lambda$, the empty string; then $\sigma \in \Sigma_{\lambda,s}$ denotes a constant of sort $s$. \[\] Regular signatures allow us to define the least sort of a term, and to give an order-sorted generalization of the usual term algebra construction. Intuitively, regularity means that the set of instances of an overloaded operator $\sigma$ whose arities are bounded below (by $w0$) has a member with least arity and least sort.

**Definition 2:** An order-sorted signature $\Sigma$ is regular iff given $w0\leq w1$ in $S*$ and $\sigma \in \Sigma_{w1,s1}$ there is a least $\langle w, s \rangle \in S^* \times S$ such that $\sigma \in \Sigma_{w,s}$ and $w0 \leq w$. A regular signature is called coherent if each connected component of its sort set is filtered\(^6\). \[\]

**Definition 3:** Let $(S,\leq,\Sigma)$ be an order-sorted signature. Then an $(S,\leq,\Sigma)$-algebra is an $(S,\Sigma)$-algebra $A$ such that

1. $s < s'$ in $S$ implies $A_s \subseteq A_{s'}$ and
2. $\sigma \in \Sigma_{w1,s1} \cap \Sigma_{w2,s2}$ with $s1 \leq s2$ and $w1 \leq w2$ implies $A_\sigma: A_{w1} \rightarrow A_{s1}$ equals $A_\sigma: A_{w2} \rightarrow A_{s2}$ on $A_{w1}$.

When $(S,\leq)$ is clear, $(S,\leq,\Sigma)$-algebras may be called order-sorted $\Sigma$-algebras. We may also write $A_\omega^w$ instead of $A_\sigma: A_{w} \rightarrow A_{s}$. \[\]

**Definition 4:** Let $(S,\leq,\Sigma)$ be an order-sorted signature, and let $A$ and $B$ be order-sorted $(S,\leq,\Sigma)$-algebras. Then a $(S,\leq,\Sigma)$-homomorphism $h: A \rightarrow B$ is a $(S,\Sigma)$-homomorphism satisfying the following restriction condition

$s < s'$ and $a \in A_s$ imply $h_s(a) = h_{s'}(a)$.

When the partially ordered set $(S,\leq)$ is clear, $(S,\leq,\Sigma)$-homomorphisms are also called order-sorted $\Sigma$-homomorphisms. The $(S,\leq,\Sigma)$-algebras and their $(S,\leq,\Sigma)$-homomorphisms form a category that we denote $\text{OSAlg}_\Sigma$. \[\]

**OSA strictly generalizes MSA.** In the sense that any many-sorted $(S,\Sigma)$-algebra is an order-sorted $(S,\leq,\Sigma)$-algebra for $\leq$ the trivial ordering on $S$, with $ss'$ iff $s=s'$; then $\text{OSAlg}_\Sigma \rightarrow \text{Alg}_\Sigma$ and the forgetful functor $\text{OSAlg}_\Sigma \rightarrow \text{Alg}_\Sigma$ is the identity. Similarly, the rules of OSA deduction given later specialize to MSA rules for this ordering.

Given an order-sorted signature $\Sigma$, we construct the order-sorted $\Sigma$-term algebra $\mathcal{T}_\Sigma$ as the least family $\{\mathcal{T}_{\Sigma,s} \mid s \in S\}$ of sets satisfying the following conditions (note our somewhat pedantic, but temporary, use of $\mathcal{T}$ and $\mathcal{S}$ for parentheses as formal syntactic symbols):

- $\Sigma_{h,s} \subseteq \mathcal{T}_{\Sigma,s}$ for $s \in S$;
- $\mathcal{T}_{\Sigma,s} \subseteq \mathcal{T}_{\Sigma,s'}$ if $s \leq s'$;

\[^6\text{Given a poset } (S,\leq), \text{let } = \text{ denote the transitive and symmetric closure of } \leq. \text{ Then } = \text{ is an equivalence relation whose equivalence classes are the connected components of } (S,\leq); \text{ a connected component is filtered iff whenever } s \text{ and } s' \text{ are in it, then there is an } s'' \text{ in it with } s'' \preceq s \text{ and } s'' \preceq s'.\]
These terms are ground terms, i.e., they involve no variables. Terms with variables arise from ground terms by enlarging the signature \( \Sigma \) with an \( S \)-sorted set \( X \) of additional constants for the variables satisfying \( X_s \cap X_{s'} = \emptyset \) when \( s \neq s' \). This gives an extended signature \( \Sigma(X) \) and an algebra \( T^\Sigma(X) \) which is denoted \( T^\Sigma(X) \) when viewed as a \( \Sigma \)-algebra. Let \( \text{vars}(t) \) denote the \( S \)-sorted set of variables that occur in \( t \); if \( t \in T^\Sigma(X) \), then \( \text{vars}(t) \subseteq X \).

Although a given term in \( T^\Sigma \) may have many different sorts, we still have

**Proposition 5:** If \( \Sigma \) is regular, each term \( t \) in \( T^\Sigma \) has a least sort, denoted \( LS(t) \).

**Definition 6:** Let \( \Sigma \) be an order-sorted signature. Then an order-sorted \( \Sigma \)-algebra \( I \) is initial in the class of all order-sorted \( \Sigma \)-algebras iff there is a unique order-sorted \( \Sigma \)-homomorphism from \( I \) to any other order-sorted \( \Sigma \)-algebra.

**Theorem 7:** Let \( \Sigma \) be a regular order-sorted signature. Then \( T^\Sigma \) is an initial order-sorted \( \Sigma \)-algebra.

**Corollary 8:** Let \( A \) be an order-sorted \( \Sigma \)-algebra and let \( f: X \rightarrow A \) be an \( S \)-sorted function (with \( X \) disjoint from \( \Sigma \)); such a function is called an assignment from \( X \) to \( A \). Then, by initiality, there is a unique order-sorted \( \Sigma \)-homomorphism \( f*: T^\Sigma(X) \rightarrow A \) that extends \( f \).

This is usually expressed by saying that \( T^\Sigma(X) \) is the free order-sorted \( \Sigma \)-algebra on \( X \).

Given an MSA equation \( t = t' \), we are forced to require that \( t \) and \( t' \) have the same sort, but OSA allows more flexibility. Since in a coherent signature two sorts in the same connected component always have a common supersort, it is enough to require that the sorts of \( t \) and \( t' \) lie in the same connected component.

**Definition 9:** For \( (S, \leq, \Sigma) \) a coherent order-sorted signature, a \( \Sigma \)-equation is a triple \( (X, t, t') \), where \( X \) is an \( S \)-sorted set of variable symbols, and \( t, t' \) are in \( T^\Sigma(X) \) with \( LS(t) \) and \( LS(t') \) in the same connected component of \( (S, \leq) \); we may write \( (\forall X) t = t' \). An order-sorted \( \Sigma \)-algebra \( A \) satisfies a \( \Sigma \)-equation \( (\forall X) t = t' \) iff \( \gamma_{LS(t)}(t) = \gamma_{LS(t')}(t') \) in \( A \) for every assignment \( \gamma: X \rightarrow A \).

Similarly, \( A \) satisfies a set \( \Gamma \) of \( \Sigma \)-equations iff it satisfies each member. The case of conditional equations and their satisfaction is similar.

### 3 Order-Sorted Equational Deduction

For \( (S, \leq, \Sigma) \) a coherent order-sorted signature and \( X, Y \) two \( S \)-sorted variable sets, a substitution is an \( S \)-sorted map \( \theta: X \rightarrow T^\Sigma(Y) \), i.e., if \( x \) has sort \( s \) then \( \theta(x) \in T^\Sigma(Y)_s \); this is a special assignment, where the values of variables lie in a term algebra. The unique order-sorted \( \Sigma \)-homomorphism \( \theta*: T^\Sigma(X) \rightarrow T^\Sigma(Y) \) induced by \( \theta \) will often be denoted \( \theta \) also. Notice that we part company here with those who define a substitution to be an endomorphism of the
terms over a fixed (usually infinite) set of variables; the trouble with such a definition is that it allows too much scope for variation on variables that are not actually used in some given problem, and for the variables that are used it is inflexible in not allowing their sorts to move up and down in the sort hierarchy. The inadequacies of the traditional approach are further discussed in Section 6.

Given an order-sorted signature $\Sigma$ and a set $\Gamma$ of conditional $\Sigma$-equations, the following are the rules for deriving (unconditional) equations:

1. **Reflexivity.** Each equation $(\forall X) t=t$ is derivable.
2. **Symmetry.** If $(\forall X) t=t'$ is derivable, then so is $(\forall X) t'=t$.
3. **Transitivity.** If the equations $(\forall X) t=t', (\forall X) t'=t''$ are derivable, then so is $(\forall X) t=t''$.
4. **Congruence.** Given $t \in T_\Sigma(X)$ and substitutions $\theta, \theta' : X \rightarrow T_\Sigma(Y)$ such that for each $x$ in $X$, the equation $(\forall Y) \theta(x) = \theta'(x)$ is derivable, then $(\forall Y) \theta(t) = \theta'(t)$ is also derivable.
5. **Substitutivity.** Given a conditional equation $(\forall X) t_1 = t_2$ if $C$ in $\Gamma$, if $\theta : X \rightarrow T_\Sigma(Y)$ is a substitution such that for each $u=v$ in $C$ the equation $(\forall Y) \theta(u) = \theta(v)$ is derivable, then so is $(\forall Y) \theta(t_1) = \theta(t_2)$.

When the equations in $\Gamma$ are unconditional, rule (5) becomes

(5') **Unconditional Substitutivity.** If $(\forall X) t_1 = t_2$ is in $\Gamma$, and if $\theta : X \rightarrow T_\Sigma(Y)$ is a substitution, then $(\forall Y) \theta(t_1) = \theta(t_2)$ is derivable.

**Theorem 10:** Completeness Theorem: [Goguen & Meseguer 88b] For an order-sorted signature $\Sigma$ and a set $\Gamma$ of conditional $\Sigma$-equations, the following are equivalent:

- $(\forall X) t=t'$ is derivable from $\Gamma$ using rules (1)-(5).
- $(\forall X) t=t'$ is satisfied by every order-sorted $\Sigma$-algebra that satisfies $\Gamma$.

When the equations in $\Gamma$ are unconditional, the same holds replacing rule (5) by (5').

Derivability from $\Gamma$ using these rules of deduction defines a congruence relation on $T_\Sigma(X)$; let $T_{\Sigma,\Gamma}(X)$ denote the quotient by it.

**Corollary 11:** Initiality Theorem: [Goguen & Meseguer 88b] For $\Sigma$ a coherent order-sorted signature and $\Gamma$ a set of conditional $\Sigma$-equations, $T_{\Sigma,\Gamma}(\emptyset)$ (henceforth denoted $T_{\Sigma,\Gamma}$) is an initial $(\Sigma, \Gamma)$-algebra and $T_{\Sigma,\Gamma}(X)$ is a free $(\Sigma, \Gamma)$-algebra on $X$.

**4 Most General Axioms**

Since $E$-unification algorithms can be quite involved, it is useful to investigate conditions under which existing unsorted $E$-unification algorithms can be reused and knowledge can be transferred to solve order-sorted unification problems. We show below that such reuse is possible if the axioms in $E$ are "most general." Intuitively, most general axioms are sort-preserving and are not restricted in application by the sort of their variables. Of course, $E$-unification for axioms $E$ not satisfying this condition is also quite interesting, but it is outside the scope of this paper. Although conceptually we will for the most part reason by
factoring order-sorted unification problems into an unsorted phase followed by an order-sorted phase, we do not wish to suggest that in practice this is the most efficient way to implement an algorithm. The quasi-linear order-sorted unification algorithm that we present in Section 8 incorporates order-sorted type checking inside the unification process. This is in general a more efficient strategy leading to early detection of failures and allowing drastic reductions in the search space.

Given an order-sorted signature \( \Sigma \), the unsorted signature, denoted \( I_\Sigma \), has the same function symbols as \( \Sigma \) with \( \sigma \) of arity \( n \) in \( I_\Sigma \) iff there are sorts \( s_1, \ldots, s_n, s \) in \( S \) such that \( \sigma : s_1 \ldots s_n \rightarrow s \) is in \( \Sigma \). \( T_{I_\Sigma}(X) \) will denote the unsorted \( I_\Sigma \)-term algebra on an (unsorted) set of variables \( X \); similarly, \( T_{I_\Sigma,\Gamma}(X) \) will denote the free algebra on the variables \( X \) in the class of all \( I_\Sigma \)-algebras that satisfy some (unsorted) equations \( \Gamma \).

In the rest of this paper, it will often be convenient to write \( S \)-sorted sets of variables in the form \( X^p \) where \( X \) is an unsorted set of variables and \( \rho : X \rightarrow S \) is a map called a sort assignment. Given an ordering \( \leq \) on \( S \), sort assignments inherit an ordering by \( \rho \preceq \rho' \) iff \( \rho(x) \leq \rho'(x) \) for each \( x \) in \( X \). If \( \rho \preceq \rho' \) we then say that \( \rho \) is a specialization of \( \rho' \). Let \( \text{Spec}(X) \) denote the poset of all sort assignments of \( X \) for a given poset \((S, \leq)\). For an \( S \)-sorted set, \( I_A \) will denote the set \( I_A = \cup_{s \in S} A_s \). For \( \Sigma \) an order-sorted signature and \( \rho \) a sort assignment, we always have \( I_{\Sigma_\rho}(X^p) \subseteq T_{\Sigma}(X) \). A \( \Sigma \)-term \( t \) is \( \Sigma \)-sensible iff \( t \in T_{\Sigma_\rho}(X^p) \) for some sort assignment \( \rho \), when \( X = \text{vars}(t) \). To emphasize the dependence on a sort assignment \( \rho \), we shall let \( \text{L}_S^\rho(t) \) denote the least sort of a term \( t \) when its variables have the sorts given by \( \rho \).

**Definition 12:** Let \( t \) be a \( \Sigma \)-sensible term and let \( X \) be the set of variables occurring in \( t \). Then a profile for \( t \) is a family \( \{ \rho_j \mid j \in J \} \) of sort assignments of \( X \) such that \( t \in T_{\Sigma_\rho}(X^p) \) for each \( j \in J \), and such that for any sort assignment \( \rho \) with \( t \in T_{\Sigma_\rho}(X^p) \) there is an index \( j \in J \) such that \( \rho_j \preceq \rho \) .

**Definition 13:** For \( \Sigma \) a coherent order-sorted signature, a set of \( \Sigma \)-equations \( E \) is most general iff it can be decomposed as a union \( E = \cup_{i \in I} E_i \) where for each \( i \in I \), \( E_i = \{( \forall x_i \rho_j \mid j \in J_i \} \) with

1. \( \text{vars}(t_i) = \text{vars}(t'_i) \),
2. \( \{ \rho_j \mid j \in J_i \} \) is a profile for both \( t_i \) and \( t'_i \), and
3. \( \text{LS}(t_i) = \text{LS}(t'_i) \), for any \( \rho \) such that \( t_i t'_i \in T_{\Sigma_\rho}(X^p) \).

Our interest in most general axioms is justified by

**Theorem 14:** For \( \Sigma \) a coherent signature and \( E \) a set of most general \( \Sigma \)-axioms, letting \( I_\Sigma \) denote the \( I_\Sigma \)-equations obtained from \( E \) by forgetting the quantifiers, we have for any set of variables \( X \) and sort assignment \( \rho \), that \( I_{\Sigma_\rho}(X^p) \subseteq T_{\Sigma_\rho,\Gamma}(X) \). Moreover, whenever \( t \in T_{\Sigma_\rho}(X^p) \) and \( t=t' \) is provable from \( I_\Sigma \) by the rules of (unsorted) equational deduction, then \( t' \in T_{\Sigma_\rho}(X^p) \), \( \text{vars}(t) = \text{vars}(t') \), \( \text{LS}(t) = \text{LS}(t') \), and \( (\forall X \rho_0) \ t=t' \) is provable from \( E \) by the rules of order-sorted equational deduction. In particular, the functions \( \text{LS} \) and \( \text{vars} \) do not depend on representatives and can be extended to equivalence classes by \( \text{LS}([t]) = \text{LS}(t) \) and \( \text{vars}([t]) = \text{vars}(t) \).

**Proof:** The rules of order-sorted equational deduction are entirely similar to the rules for
unsorted deduction. Indeed, forgetting about the quantifiers and making the signature and
the substitutions unsorted, rules (1)-(5') yield corresponding sound and complete rules for
unsorted deduction. This means that if an equation is provable by order-sorted deduction, then
it is also provable by unsorted deduction. The converse does not hold in general, since the
sorts of the quantified variables have to be preserved by order-sorted substitutions, and this
imposes further restrictions. However, under the theorem's assumptions a converse holds in
the very strong sense that if \( t \in T(X^P) \) and \( t=t' \) is provable from \( \text{IEI} \) by the rules of (unsorted)
equational deduction, then \( t' \in T(X^P) \), \( \text{vars}(t)=\text{vars}(t') \), \( \text{LS}(t)=\text{LS}(t') \), and \( (\forall X^P) t=t' \) is
provable from \( E \) by the rules of order-sorted equational deduction. By induction, the result
reduces to proving that if \( t \in T(X^P) \) and \( t=t' \) is one step of unsorted replacement of equals by
equals using \( \text{IEI} \), the claim holds. Therefore, there is an equation \( t_i=t'_i \) in \( \text{IEI} \) and an occurrence
\( \alpha \) in \( t \) such that the subterm \( t/\alpha \) at that occurrence is of the form \( \alpha x(t_i) \) for \( \alpha \) an unsorted
substitution such that \( t' \) is the result of replacing \( t(x(t_i)) \) at occurrence \( \alpha \) in \( t \), which we
denote by \( t'=t'(\alpha \leftarrow x(t_i)) \). The first thing to realize is that there is an \( S \)-sorted substitution \( \tau \) such that \( \tau(t_i)=t(x(t_i)) \) and \( \tau(t'_i)=t'(x(t'_i)) \). Indeed, let \( X_t=\{x_1, \ldots, x_n\} \) be the variables in \( t_i \) and let
\( u_i=\text{ls}(x_i) \). Since \( t \) is a well-formed order-sorted term, so too are its subterms. Let \( s_i=\text{LS}_P(u_i) \) and let \( v \) be the sort assignment mapping \( x_i \) to \( s_i \). We have \( t_i, t'_i \in T(X^P) \) and therefore there is a sort assignment \( \rho_{ij} \) in the profile of \( t_i, t'_i \) such that \( \rho_{ij} \geq v \). Therefore, there is an \( S \)-sorted
substitution \( \tau: X_t \rho_i \rightarrow T(X^P) \) whose unsorted version is \( \text{lt} \). By order-sorted unconditional
substitutivity, this proves that \( (\forall X^P) t/\alpha=t'/\alpha \) is provable from \( E \). Since the equations \( E \) are
general, it is easy to check that \( \text{LS}(t/\alpha)=\text{LS}(t'/\alpha) \), and therefore \( t' \in T(X^P) \). Then, by
considering the term \( t(\alpha \leftarrow z) \) with \( z \) a new variable of sort \( \text{LS}(t/\alpha) \) and applying to this term the
rule of order-sorted congruence with two substitutions mapping \( z \) to \( t/\alpha \) and \( t'/\alpha \) and leaving
the rest unchanged, we obtain the desired result that \( (\forall X^P) t=t' \) is provable from \( E \) by the rules
of order-sorted equational deduction. Checking that \( \text{vars}(t)=\text{vars}(t') \) and that \( \text{LS}(t)=\text{LS}(t') \) is
now trivial. \( \Box \)

5 Order-Sorted Unifiers and E-Unifiers

We first motivate a very general notion of system of equations in an order-sorted context.
Roughly, we may want to solve a system of equations, say \( t_1=t'_1, \ldots, t_n=t'_n \) where the \( t_i \) and \( t'_i \)
are \( \Sigma \)-sensible terms, but initially we impose no constraints on the variables of the terms, and
part of the answer should then be what sorts the variables must have in order for a solution to
exist. In other contexts, however, we may want to initially constrain the sorts of the variables,
or at least some of them, to specific sorts. We can formalize this in a general way by assuming
that part of the initial conditions of our system of equations is a partial function \( \rho_0: X \rightarrow S \)
where \( X \) is the set of all variables occurring in some equation; the case where no constraints
are initially placed on the sorts then corresponds to giving the empty function.

Definition 15: For \( \Sigma \) a coherent signature, a system of equations is a pair \( (\Gamma, \rho_0) \) where \( \Gamma \) is a
finite set of \( \Sigma \)-sensible (unquantified) \( \text{IEI} \)-equations and \( \rho_0: X \rightarrow S \) is a partial function with
\( X=\text{vars}(\Gamma)=\bigcup \{\text{vars}(t) \mid \exists t \text{ or } t'=t \text{ in } \Gamma\}. \( \Box \)
We next need to explain what it means for a system of equations to be solvable in a free order-sorted algebra or in a free algebra in a class of order-sorted algebras satisfying some axioms E. This leads to the concept of order-sorted unifiers and E-unifiers. Since we will find it useful to relate order-sorted unifiers to unsorted unifiers, recall first of all that an unsorted \(\Sigma\)-unifier for a set \(\Gamma\) of \(\Sigma\)-equations is just a map \(\theta: X \rightarrow T_{\Sigma}(Y)\), called a substitution, such that \(\theta(t) = \theta(t')\) for each \(t = t'\) in \(\Gamma\) (where we denote also by \(\theta\) the unique extension of \(\theta\) to a \(\Sigma\)-homomorphism \(T_{\Sigma}(X) \rightarrow T_{\Sigma}(Y)\)). Similarly, an unsorted \(\Sigma, E\)-unifier is a map \(\theta: X \rightarrow T_{\Sigma, E}(Y)\) such that \(\theta(t) = \theta(t')\) (where now we denote by \(\theta\) the unique extension of \(\theta\) to a \(\Sigma\)-homomorphism \(T_{\Sigma}(X) \rightarrow T_{\Sigma, E}(Y)\)).

**Definition 16:** For \(\Sigma\) a coherent signature, a \(\Sigma\)-unifier for a system of equations \((\Gamma, \rho_0)\) is a \(\Sigma\)-unifier \(\theta: X \rightarrow T_{\Sigma}(Y)\) for \(\Gamma\) such that there is a sort assignment \(\rho: Y \rightarrow S\) satisfying

1. for each \(t = t'\) in \(\Gamma\), \(\theta(t) = \theta(t') \in T_{\Sigma}(Y^0)\), and
2. \(LS(\theta(x)) < s\) whenever \(\rho(x) = s\) (note that \(\theta(x) \in T_{\Sigma}(Y^0)\) since it is a subterm of a term \(\theta(t)\) for some term \(t\) appearing in some equation in \(\Gamma\)).

Note that our \(\Sigma\)-unifier \(\theta\) can therefore be expressed as a function \(\theta: X \rightarrow T_{\Sigma}(Y^0)\). We will favor this later notation over the less informative \(\theta: X \rightarrow T_{\Sigma}(Y)\) since it has the advantage of making the sort assignment \(\rho\) explicit.

For \(E\) a set of \(\Sigma\)-equations, we say that a substitution \(\theta': X \rightarrow T_{\Sigma, E}(Y^0)\) is a presentation of a function \(\theta: X \rightarrow T_{\Sigma, E}(Y^0)\) iff \(\theta'(x) = \theta(x)\) for each \(x\) in \(X\). We say that the map \(\theta: X \rightarrow T_{\Sigma, E}(Y^0)\) is a \(\Sigma, E\)-unifier for \((\Gamma, \rho_0)\) iff it has a presentation \(\theta': X \rightarrow T_{\Sigma, E}(Y^0)\) such that

1. for each \(t = t'\) in \(\Gamma\), \(\theta'(t) = \theta'(t') \in T_{\Sigma, E}(Y^0)\) and \([\theta'(x)] = [\theta'(x')]\), and
2. \(LS(\theta'(x)) < s\) whenever \(\rho(0(x)) = s\).

The presentation \(\theta'\) is then called admissible for \((\Gamma, \rho_0)\). Note that for \(E = \emptyset\) a \(\Sigma, E\)-unifier is exactly the same as a \(\Sigma\)-unifier.

**Remark 17:** The requirement that the signature \(\Sigma\) be coherent is not superfluous. Without it, one can have equations that can be solved in a free order-sorted algebra but cannot be solved in another algebra isomorphic to it (and therefore also free). Take \(\Sigma\) with \(S\) consisting of three sorts \(A, B, C\), each with a constant, \(a\) of sort \(A\), \(b\) of sort \(B\), \(c\) of sort \(C\), and with \(B \prec A, B \prec C\). The equation \(a = c\) cannot be solved in \(T_{\Sigma}\) but it can be solved in the algebra \(M\) with \(M_A = \{b, d\}, M_B = \{b\}, M_C = \{b, d\}\) with \(a\) and \(c\) interpreted by \(d\). Indeed, \(M\) and \(T_{\Sigma}\) are isomorphic.

Most general \(\Sigma\)-axioms permit a simple way of characterizing \(\Sigma, E\)-unifiers:

**Lemma 18:** For \(\Sigma\) a coherent signature and for \(E\) a set of most general \(\Sigma\)-axioms, a map \(\theta: X \rightarrow T_{\Sigma, E}(Y^0)\) is a \(\Sigma, E\)-unifier for \((\Gamma, \rho_0)\) iff

1. \(\theta\) is a \(\Sigma, E\)-unifier for \(\Gamma\),
2. \(\theta(t), \theta(t') \in T_{\Sigma, E}(Y^0)\) for each \(t = t'\) in \(\Gamma\), and
3. \(LS(\theta(x)) < s\) whenever \(\rho(0(x)) = s\).
6 Coverings and Initial Families of Unifiers

The usual story about unifiers is told in terms of endomorphisms of a free algebra with infinitely many variables by defining a subsumption preorder among such substitutions; then one looks for families of unifiers that generate the ideal of all unifiers for a system of equations, etc. Although this tale has some syntactic advantages and a venerable tradition, it has some serious disadvantages:

1. It obscures the model-theoretic semantics of unification and, in particular, it hinders the understanding of unification as a generalized initiality problem.

2. For order-sorted unification, given that many different sorts can be assigned to the same variable, attempts at formulating the problem via a monolithic free algebra with infinitely many variables can be awkward and run the risk of being conceptually misleading.

3. Many-sorted and order-sorted algebras may have some of their sorts empty; unifiers landing on a free algebra with too many variables may fail to cover some unifiers, because no homomorphisms can be defined to unifiers landing on free algebras having some of their sorts empty; besides, the map $\sigma_{\Sigma_E}(Y) \to \sigma_{\Sigma_E}(X)$ associated to an $S$-sorted inclusion $Y \subseteq X$ is in general not injective [Goguen & Meseguer 85].

In our opinion, the clearest way to understand unifiers is as forming a category. In the unsorted case one might deem this approach uneconomical; however, we think that the investment more than pays for itself in the many-sorted and order-sorted cases, and that it is enlightening even in the unsorted case.

We now develop some categorical concepts that will later be applied to unifiers. Let $C$ be an arbitrary category; we can give a preorder to the objects of $C$ by defining $X \succ Y$ iff there is a map $X \to Y$ in $C$. We say that a family $G$ of objects is a covering iff for any object $Y$ in $C$ there is an object $X$ in $C$ such that $X \succ Y$. A covering $M$ is called minimal iff it satisfies (either of)

the following equivalent conditions:

- For any $X$ in $C$, $M \setminus \{X\}$ is not a covering.
- Whenever $X \to X'$ in $C$ with $X$ and $X'$ in $M$, then $X = X'$.

A minimal covering consisting of exactly one object is called a most general object in $C$; of course, this is just an object $X$ such that $X \succ Y$ for each $Y$ in $C$. Any category $C$ has a covering;
for instance, the family of all objects in \( C \) is a covering. Also, if \( C \) has a finite covering, then it has a minimal (and finite) covering. In general, however, \( C \) may fail to have a minimal covering. When a minimal covering does exist, we have the following:

**Lemma 19:** Let \( \mathcal{G} \) be a covering and \( \mathcal{M} \) a minimal covering in \( C \). Then, there is a minimal covering \( \mathcal{M}' \subseteq \mathcal{G} \) and a unique bijection \( F: \mathcal{M} \to \mathcal{M}' \) such that for each \( X \) in \( \mathcal{M} \), \( X \leq F(X) \) and \( F(X) \geq X \).

**Proof:** Since \( \mathcal{G} \) is a covering, we can choose for each \( X \) in \( \mathcal{M} \) an object \( F(X) \) in \( \mathcal{G} \) such that \( F(X) \geq X \). Let \( \mathcal{M}' = \{ F(X) | X \in \mathcal{M} \} \). This defines a surjective function \( F: \mathcal{M} \to \mathcal{M}' \). Since \( \mathcal{M}' \) is a covering, so is \( \mathcal{M}' \). To see that \( F \) is injective and that \( \mathcal{M}' \) is minimal, let \( F(X_1) \to F(X_2) \) in \( \mathcal{C} \), since \( \mathcal{M} \) is a covering, there is an object \( X_3 \) in \( \mathcal{M} \) with \( X_3 \to F(X_1) \). Then, minimality of \( \mathcal{M} \) forces \( X_1 = X_2 = X_3 \). This also shows that \( X \geq F(X) \). The map \( F \) is unique since whenever \( X_1 \to F(X_2) \), then \( X_1 \to F(X_2) \to X_2 \) and \( X_1 = X_2 \). □

A covering \( I \) is called an **initial family** iff for each object \( Y \) and each \( X \) in \( I \) there is at most one map \( X \to Y \). An initial family \( I \) is **minimal** iff it is minimal as a covering. We say that \( C \) has an **initial object** iff it has a minimal initial family consisting of a single object. Generalizing the uniqueness of initial objects up to isomorphism, we also have the uniqueness of minimal initial families up to isomorphism:

**Fact 20:** If \( I \) and \( I' \) are minimal initial families in \( C \), then there exists a unique bijection \( F: I \to I' \) such that each \( X \) in \( I \) is isomorphic to \( F(X) \) in \( I' \). □

Let us now apply this to unification. For \( \Sigma \) an unsorted signature, and for \( \Gamma \) and \( \Gamma' \) sets of (unquantified) \( \Sigma \)-equations, we let the category \( \text{Un}_{\Sigma,\Gamma}(\Gamma) \) have objects \( \Sigma,\Gamma \)-unifiers for \( \Gamma \); and given \( \theta: X \to T_{\Sigma,\Gamma}(Y) \) and \( \eta: X \to T_{\Sigma,\Gamma}(Z) \) in \( \text{Un}_{\Sigma,\Gamma}(\Gamma) \) a map from \( \theta \) to \( \eta \) is just a \( \Sigma \)-homomorphism \( \mu: T_{\Sigma,\Gamma}(Y) \to T_{\Sigma,\Gamma}(Z) \) such that \( \mu \circ \theta = \eta \). By \( \text{Un}_{\Sigma,\Gamma}(\Gamma) \) we just mean \( \text{Un}_{\Sigma,\Gamma}(\Gamma) \). For \( \Sigma \) a coherent order-sorted signature, for \( E \) a set of most general \( \Sigma \)-axioms and \( (\Gamma,\rho_0) \) a system of \( \Sigma \)-equations, the category \( \text{Un}_{\Sigma,\Gamma}(\Gamma,\rho_0) \) has as objects pairs \((\theta, T_{\Sigma,\Gamma}(Y))\) with \( \theta: X \to T_{\Sigma,\Gamma}(Y) \) a \( \Sigma \)-E-unifier for \((\Gamma,\rho_0)\); given \((\theta, T_{\Sigma,\Gamma}(Y))\) and \((\eta, T_{\Sigma,\Gamma}(Z))\) in \( \text{Un}_{\Sigma,\Gamma}(\Gamma,\rho_0) \), a map \( \mu: (\theta, T_{\Sigma,\Gamma}(Y)) \to (\eta, T_{\Sigma,\Gamma}(Z)) \) is just a \( \Sigma \)-homomorphism \( \mu: T_{\Sigma,\Gamma}(Y) \to T_{\Sigma,\Gamma}(Z) \) such that \( \mu \circ (\theta(x)) = (\eta(x)) \) for each \( x \) in \( X \). By \( \text{Un}_{\Sigma,\Gamma}(\Gamma,\rho_0) \) we just mean \( \text{Un}_{\Sigma,\Gamma}(\Gamma,\rho_0) \). Both in \( \text{Un}_{\Sigma,\Gamma}(\Gamma) \) and \( \text{Un}_{\Sigma,\Gamma}(\Gamma,\rho_0) \), a covering (or minimal covering, or most general object, or initial family) will be called a covering (or minimal covering, or most general object, or initial family) of unifiers.

A \( \Sigma,\Gamma \)-E-unifier \( \theta: X \to T_{\Sigma,\Gamma}(Y) \) is called **sober** iff there is a presentation \( \theta': X \to T_{\Sigma}(Y) \) of \( \theta \) such that \( Y = \bigcup_{x \in X} \vars(\theta'(x)) \). Similarly, For \( \Sigma \) a coherent signature, \( E \) a set of most general \( \Sigma \)-axioms, and \( (\Gamma,\rho_0) \) a system of \( \Sigma \)-equations, \( (\theta, T_{\Sigma,\Gamma}(Y)) \) in \( \text{Un}_{\Sigma,\Gamma}(\Gamma,\rho_0) \) is **sober** iff \( Y = \bigcup_{x \in X} \vars(\theta(x)) \) (note that, by the equations \( E \) being most general, \( \vars \) is defined on entire \( E \)-equivalence classes). We can always assume that coverings of unifiers are sober (i.e., all their members are sober), thanks to the following:

**Lemma 21:** \( \text{Un}_{\Sigma,\Gamma}(\Gamma) \) has a sober covering. For \( \Sigma \) a coherent signature, \( E \) a set of most
general $\Sigma$-axioms, and $(\Gamma, \rho_0)$ a system of $\Sigma$-equations, $Un_{\Sigma,E}(\Gamma, \rho_0)$ has a sober covering.

**Proof:** We prove the order-sorted case. Let $g$ be a covering (since one always exists) for $Un_{\Sigma,E}(\Gamma, \rho_0)$, and form $g'$ by replacing each $(\theta, T_{\Sigma,E}(Y))$ in $g$ by $(\gamma, T_{\Sigma,E}(vars(codomain(\theta)))$ with $vars(codomain(\theta))=\cup_{x} x vars(\theta(x))$ (note that, by Theorem 14 vars is defined on $E$-equivalence classes) and if $\theta(x)=[u_x]$ then $\gamma(x)=[u_x]$ for each $x$ in $X$. Then, the $\Sigma$-homomorphism $\gamma_{\Sigma,E}(vars(codomain(\theta))) \rightarrow T_{\Sigma,E}(Y)$ induced by the $S$-sorted inclusion $vars(codomain(\theta)) \subseteq Y$ makes $(\gamma, T_{\Sigma,E}(vars(codomain(\theta))) \rightarrow (\theta, T_{\Sigma,E}(Y)))$ and therefore $g'$ is a covering, and sober by construction.

**Theorem 22:** For $\Sigma$ a coherent signature, $E$ a set of most general $\Sigma$-axioms, and $(\Gamma, \rho_0)$ a system of $\Sigma$-equations, if $g$ is a sober covering of $\Sigma,E$-unifiers in $Un_{\Sigma,E}(\Gamma, \rho_0)$ then the family $\{ (\theta, T_{\Sigma,E}(Y)) \in Un_{\Sigma,E}(\Gamma, \rho_0) | \theta \in g \}$ is a sober covering of $\Sigma$-unifiers in $Un_{\Sigma,E}(\Gamma, \rho_0)$.

**Proof:** Regarding sobriety, there is nothing to check. Let $(\eta, T_{\Sigma,E}(Z^\theta)) \in Un_{\Sigma,E}(\Gamma, \rho_0)$ and let $\theta \in g$ with $\theta \eta$, say, $\theta: X \rightarrow T_{\Sigma,E}(Y)$ and $\mu: T_{\Sigma,E}(Y) \rightarrow T_{\Sigma,E}(Z)$ with $\mu \eta = \eta$. Let $\theta(x)=[u_x]$, $\eta(x)=[v_x]$, $\mu(y)=[w_y]$. By sobriety and the axioms $E$ being most general, we have $Y=\cup_{x} x vars(u_x)$. By $\mu$ being a $\Sigma$-homomorphism we have $v_y$ provably equal by unsorted equational deduction using $E$ to the term $u_y(y_1 =_w y_1, \ldots, y_n =_w y_n)$ for $\{y_1, \ldots, y_n\}=vars(u_y)$. Since the axioms $E$ are most general and $v_x \in T_{\Sigma,E}(Z^\theta)$, we have

$$u_y(y_1 =_w y_1, \ldots, y_n =_w y_n) \in T_{\Sigma,E}(Z^\theta)$$

and therefore, by $\theta$ sober, $w_y \in T_{\Sigma,E}(Z^\theta)$ for each $y$ in $Y$. Defining the sort assignment $\rho: Y \rightarrow S$ by $p(y)=LS(w_y)$ (note that, since the equations $E$ are most general, $LS$ is defined on entire $E$-equivalence classes) makes the map $y \mapsto [w_y]$ $S$-sorted and therefore there is a $\Sigma$-homomorphism $\mu$ such that $(\theta, T_{\Sigma,E}(Y)) \rightarrow (\eta, T_{\Sigma,E}(Z^\theta))$.

**Corollary 23:** For $\Sigma$ a coherent signature, and $(\Gamma, \rho_0)$ a system of $\Sigma$-equations, any sober covering of $Un_{\Sigma,E}(\Gamma, \rho_0)$ is an initial family.

**Proof:** Specializing the reasoning in the proof of the above theorem to the case $E=\emptyset$, we get $u_y(y_1 =_w y_1, \ldots, y_n =_w y_n) = v_y$ and therefore, in order for $\mu$ to be a map $(\theta, T_{\Sigma,E}(Y)) \rightarrow (\eta, T_{\Sigma,E}(Z^\theta))$ it has necessarily to send $y$ to $w_y$ and is therefore unique.

**Remark 24:** It is not hard to give examples of sober coverings of unifiers that are not initial when the most general axioms $E$ are nonempty. For example, consider the unsorted (and therefore coherent) signature $\Sigma$ consisting of unary function symbols $f,g,h,q$; let $E$ be the set of (trivially most general) axioms $\{f(k(y))=f(h(y)), g(f(x))=q(f(x))\}$, and let $\Gamma=\{g(x)=q(x)\}$. Then, the unifier $\theta: \{x \rightarrow T_{\Sigma,E}(y)\}$ mapping $x$ to $[f(y)]$ is sober and a most general unifier; however, for $\eta: \{x \rightarrow T_{\Sigma,E}(z)\}$ the unifier mapping $x$ to $[f(k(z))]$ there are two different homomorphism $\mu: [y] \mapsto [k(z)]$ and $\nu: [y] \mapsto [h(z)]$ such that $\mu \theta = \nu \eta = \eta$.

Using the above results, plus the usual Herbrand-Robinson Theorem [Herbrand 30, Robinson 65], we can get the following sharpened form of that important result:

**Corollary 25:** If $Un_{\Sigma,E}(\Gamma)$ is nonempty, then it has an initial object.
The following example illustrates the concept of order-sorted E-unifier. It also suggests that, in solving problems that involve unification, one could reduce the search space by incorporating the typechecking within the unification algorithm itself. This is actually done for syntactic unification in our Martelli-Montanari algorithm.

**Example 26:** Let the set of sorts be \( S = \{ \text{Elt}, \text{Mult} \} \) with \( \text{Elt} < \text{Mult} \). The signature \( \Sigma \) consists of a binary operation \( _-_ : \text{Mult} \times \text{Mult} \to \text{Mult} \) with the syntax of juxtaposition. The axioms \( E=AC \) are associativity and commutativity of our binary operation, and it is easy to see that they are most general. Therefore, in the free algebra modulo these axioms generated by a set \( A \) of elements of sort \( \text{Elt} \), the elements of sort \( \text{Mult} \) can be understood as the finite multisets of the set \( A \). Now consider the system \((\Gamma, \rho)\) consisting of the equation

\[
X S = Y T
\]

with \( \rho(X) = \rho(Y) = \text{Elt} \), and undefined elsewhere. This system has the following minimal covering of unsorted unifiers modulo associativity and commutativity:

1. \( X = T, Y = S \)
2. \( X = Y, S = T \)
3. \( S = (Y U), T = (X U) \)
4. \( X = (Y U), T = (S U) \)
5. \( S = (T U), Y = (X U) \)
6. \( X = (T U), Y = (S U) \)
7. \( X = (U V), S = (U' V'), Y = (V V'), T = (U U') \)

However, since \( X \) and \( Y \) must have sort \( \text{Elt} \), only the first three unifiers are possible at the order-sorted level. Indeed, letting \( \tau_i \) for \( i = 1, \ldots, 3 \) denote the first three substitutions above, \( X_1 = X_2 = (X, Y, S, T), X_3 = (X, Y, U) \), and letting \( \rho_i: X_i \to S \) for \( i = 1, \ldots, 3 \) denote the sort assignments:

1. \( \rho_1(X) = \rho_1(Y) = \rho_1(S) = \rho_1(T) = \text{Elem} \)
2. \( \rho_2(X) = \rho_2(Y) = \text{Elt}, \rho_2(S) = \rho_2(T) = \text{Mult} \)
3. \( \rho_3(X) = \rho_3(Y) = \text{Elt}, \rho_3(U) = \text{Mult} \)

we obtain a minimal covering of \( \text{Un}_{\Sigma,AC}(\Gamma, \rho) \) given by the \( (\tau_i, \Sigma_{\Sigma,AC}(X_i \rho_i)), i = 1, \ldots, 3 \).

7 Characterization Theorems

We will now give necessary and sufficient conditions for an order-sorted signature \( \Sigma \) to always have a minimal covering (or finite minimal covering, or most general unifier when solvable) of \( \Sigma, E \)-unifiers for \((\Gamma, \rho_0)\) whenever there is a minimal covering (or finite minimal covering, or most general unifier) of \( \{\Sigma, E\}-\text{unifiers for} \ \Gamma \). These results greatly generalize a characterization by [Walther 86], Thm. 6.2, in two directions: first, Walther considered only order-sorted signatures without overloading of function symbols, whereas here we consider arbitrary order-sorted signatures where the function symbols are in general overloaded; second, Walther considered only the case of syntactic unification, whereas we consider unification modulo a set of most general axioms \( E \).
We begin with the definition of a normal poset and a normal signature. Normal posets require that the set of lower bounds of a finite set of elements has a minimal covering, i.e., a "complete" collection of maximal elements. The natural numbers with their usual ordering and with an additional infinity element greater than any other number fail to be normal due to the infinity element, and even without the infinity element they are not normal (consider the lower bounds of the empty set). The concept is generalized to signatures by requiring that the sort assignments typechecking a finite family of terms have a complete collection of maximal elements.

**Definition 27:** A partially ordered set \((S,\leq)\) is called normal iff for each finite subset \(S' \subseteq S\) the poset \((\text{LBd}(S'),\leq)\) (where \(\text{LBd}(S') = \{ s \in S : \forall s' \in S' \; s' \leq s \}\)) and \(\leq\) is the restriction of the order on \(S\), considered as a category in the usual way, i.e., \(s \rightarrow s'\) if \(s \leq s'\), has a minimal covering. An order-sorted signature \((S,\leq,\Sigma)\) is called normal iff

- \((S,\leq)\) is normal, and
- for any finite family of terms \(\{t_1, \ldots, t_n\}\) (denoted \(\{t_i\}\) for short) with \(Y = \text{vars}(\{t_i\})\) and for any partial function \(\rho_0 : \{1, \ldots, n\} \rightarrow S\), the set \(\text{LBd}({\{t_i\}, \rho_0}) = \{ \rho \in \text{Spec}(Y) : \forall t_i \in T_\Sigma(Y^0), i = 1, \ldots, n \text{ and } LSp(t_i) \leq \rho_0(i) \text{ whenever } \rho_0(i) \text{ is defined} \}\) has a minimal covering.

**Theorem 28:** For \((S,\leq,\Sigma)\) a coherent order-sorted signature the following are equivalent:

1. For each set of most general \(\Sigma\)-axioms \(E\) and system of \(\Sigma\)-equations \((\Gamma, \rho_0)\) such that \(\text{Un}(\Sigma, E)(\Gamma, \rho_0)\) has a minimal covering, \(\text{Un}(\Sigma, E)(\Gamma, \rho_0)\) also has a minimal covering.
2. \((S,\leq,\Sigma)\) is normal.

**Proof:** To see that (1) \(\Rightarrow\) (2), assume that \((S,\leq,\Sigma)\) is not normal, i.e., either (i) there is a finite \(S' \subseteq S\) such that \(\text{LBd}(S')\) does not have a minimal covering or (ii) there is a finite family of terms \(\{t_i\}\) and partial function \(\rho_0\) such that \(\text{LBd}({\{t_i\}, \rho_0})\) does not have a minimal covering. In case (i), if \(S'\) is nonempty, consider the system of equations with \(\Gamma = \{x_i = x_s, s' \leq s \in S'\}\) and \(\rho_0\) the function assigning to each \(x_s\) the sort \(s\). The set of equations \(\Gamma\) has an initial unsorted (sober) unifier \(\theta\) mapping all the \(x_s\) to the same variable \(x\) and then the family \(\{(\theta, \tau_\Sigma(x_\rho_0))(x_\rho_0(x) = s, s \in \text{LBd}(S'))\}\) is an initial family of unifiers for \((\Gamma, \rho_0)\) but does not contain a minimal initial family by our assumption on \(S'\). If \(S' = \emptyset\) consider the system with \(\Gamma = \{x = y\}\) and \(\rho_0\) the undefined function. The unsorted unifier is, say the substitution mapping \(x\) to \(y\) and leaving \(y\) fixed. Any sort assignment \(\rho : \{y\} \rightarrow S\) will lead to an object of the covering in \(\text{Un}(\Sigma, E)(\Gamma, \rho_0)\) but since \(S\) does not have a minimal covering (by filteredness this means that at least one of its connected components does not have a top element), there is no minimal subcovering of unifiers. In case (ii), consider \(\Gamma = \{x_i = t_i\}\) with the \(x_i\) all new variables, and \(\rho'_0\) such that \(\rho'_0(x_i) = \rho_0(i)\) and is undefined elsewhere. \(\Gamma\) has the substitution \(x_i \mapsto t_i\) as its unsorted most general unifier. The sort assignments in \(\text{LBd}({\{t_i\}, \rho_0})\) correspond exactly to a covering of unifiers for \(\text{Un}(\Sigma, E)(\Gamma', \rho'_0)\) but, since those sort assignments do not have a minimal covering, we fail to have a minimal covering of unifiers.

The proof of (2) \(\Rightarrow\) (1) is very simple. Given a system of equations \(\Gamma = \{t_i = t'_{i} \mid i = 1, \ldots, n\}\) and \(\rho_0 : \text{vars}(\Gamma) = \{x_1, \ldots, x_m\} \rightarrow S\), we may assume that the minimal covering \(\mathcal{M}\) of unsorted \(\text{El-}\)
unifiers is sober; therefore, our task is to find a minimal subcovering of the sober covering of order-sorted unifiers \( \{ (\theta, \tau_{\Sigma, E}(Y^0)) \in \text{Un}_{\Sigma, E}(\Gamma, \rho_0) \mid \theta \in \mathcal{A} \} \). What we will do is exhibit, for each \( \theta: X \to T_{\Sigma, E}(Y) \) in \( \mathcal{A} \) presented by \( \theta': X \to T_{\Sigma, E}(Y) \), a set of maximal sort assignments \( \rho \) such that \( (\theta, \tau_{\Sigma, E}(Y^0)) \in \text{Un}_{\Sigma, E}(\Gamma, \rho_0) \). This is just the set of maximal elements of the set \( LBd(\{ u_j \mid j=1, \ldots, n+m \}, v) \) with \( u_j=\theta'(x_j) \) for \( j=1, \ldots, m \), \( u_{m+i}=\theta'(t_i) \) for \( i=1, \ldots, n \), and \( v(j)=\rho_0(x_j) \) for \( j=1, \ldots, m \) whenever \( \rho_0(x_j) \) defined and undefined otherwise.

Of course, we have to show that this is independent of the choice of presentation \( \theta' \). Since the axioms \( E \) are most general, for any sort assignment \( \rho \), if \( t \) and \( t' \) are provably equal by \( E \), \( t \in T_{\Sigma, E}(Y^0) \) iff \( t' \in T_{\Sigma, E}(Y^0) \), \( \text{vars}(t)=\text{vars}(t') \), and \( \text{LS}_\rho(t)=\text{LS}_\rho(t') \). This shows that \( LBd(\{ u_j \mid 1=j=n+m \}, v) \) above is independent of the choice of presentation.

Corollary 23 and the proof of the last theorem yield

**Corollary 29:** For \((S, \leq, \Sigma)\) a coherent order-sorted signature the following are equivalent:

1. For any system \((\Gamma, \rho_0)\) of \( \Sigma \)-equations, \( \text{Un}_{\Sigma}(\Gamma, \rho_0) \) has a minimal initial family.
2. \((S, \leq, \Sigma)\) is normal.

**Definition 30:** A partially ordered set \((S, \leq)\) is called Noetherian iff it does not have any infinite ascending chains \( s_1 < s_2 < \ldots < s_n < \ldots \). A coherent signature \((S, \leq, \Sigma)\) is called Noetherian iff \((S, \leq)\) is Noetherian.

**Corollary 31:** For \((S, \leq, \Sigma)\) a Noetherian order-sorted signature, \( E \) a set of most general \( \Sigma \)-axioms, and \((\Gamma, \rho_0)\) a system of \( \Sigma \)-equations such that \( \text{Un}_{\Sigma, E}(\Gamma) \) has a minimal covering, then \( \text{Un}_{\Sigma, E}(\Gamma, \rho_0) \) also has a minimal covering.

**Proof:** We simply have to show that any Noetherian signature is normal. Let \( S' \) be any subset of \( S \); if \( s \) in \( LBd(S') \) is not less than or equal to a maximal element in \( LBd(S') \), then we can create an infinite increasing family. The proof for \( LBd(\{ t_i \}, \rho_0) \) reduces to the observation that the poset of sort assignments \( \{ y_1, \ldots, y_n \} \to S \) is isomorphic to \( S^m \) and the remark that finite products of Noetherian posets are Noetherian.

**Definition 32:** A partially ordered set \((S, \leq)\) is called finitary iff for each finite nonempty subset \( S' \subseteq S \) the poset \( LBd(S') \leq \) has a finite and minimal covering. An order-sorted signature \((S, \leq, \Sigma)\) is called finitary\(^1\) iff

- \((S, \leq)\) is finitary, and
- for each \( \sigma \) in \( S \), \( s \) in \( S \) and \( n \) in \( \mathbb{N} \) the sets \( LBd(\sigma, s, n) = \{ s_1 \ldots s_n \mid \sigma: s_1 \ldots s_n \to s' \text{ in } \Sigma \text{ and } s' \leq s \} \) and \( LBd(\sigma, *, n) = \{ s_1 \ldots s_n \mid \sigma: s_1 \ldots s_n \to s' \text{ in } \Sigma \} \) have finite and minimal covering, denoted \( \text{max-arity}(\sigma, s, n) \), \( \text{max-arity}(\sigma, *, n) \).

\(^1\)Note that every finite order-sorted signature is finitary, but the converse does not hold. Indeed, a finitary signature may even have an infinite number of connected components.
Before stating our next characterization theorem, we develop a technical notion of multiple sort assignment that will be useful in proofs. Recall that \( \text{Spec}(X) \) was a partially ordered set. Similarly, we can give to the set of functions \( \tau : X \rightarrow \mathcal{P}(S) \), called multiple sort assignments, a quasi-order by: \( \tau \leq \tau' \) iff \( \text{LBd}(\tau(x)) \subseteq \text{LBd}(\tau'(x)) \) for each \( x \) in \( X \). This quasi-ordered set will be denoted \( \text{MSpec}(X) \). We can view partial or total functions \( X \rightarrow S \) as multiple sort assignments whose values are singletons or empty, and in this way we have inclusions

\[
\text{Spec}(X) \subseteq \text{PSpec}(X) \subseteq \text{MFInSpec}(X) \subseteq \text{MSpec}(X)
\]

and

\[
\text{Spec}(X) \subseteq \text{MFIn0Spec}(X) \subseteq \text{MFInSpec}(X) \subseteq \text{MSpec}(X),
\]

where \( \text{PSpec}(X) \) is the set of partial functions \( X \rightarrow S \), called partial sort assignments, \( \text{MFInSpec}(X) \) is the set of multiple sort assignments \( \tau \) such that \( \tau(x) \) is finite, called finitary multiple sort assignments and \( \text{MFIn0Spec}(X) \) is the set of multiple sort assignments \( \tau \) such that \( \tau(x) \) is finite and nonempty for each \( x \), called nonempty finitary multiple sort assignments.

Note that the restriction to \( \text{Spec}(X) \) of the quasi-order on \( \text{MSpec}(X) \) coincides with the original ordering given to the set \( \text{Spec}(X) \). In any quasi-ordered set we say that an element is a glb of a set \( U \) iff it is a most general element of \( \text{LBd}(U) \). The following facts will be used later:

**Lemma 33**: Let \((S, \leq)\) be a partially ordered set. Then

1. Given a family \( \{\tau_i\} \) in \( \text{MSpec}(X) \) and defining \( \cup_i \tau_i \) by \((\cup_i \tau_i)(x) = \bigcup_i (\tau_i(x))\), then \( \cup_i \tau_i \) is a glb of \( \{\tau_i\} \) in \( \text{MSpec}(X) \).
2. The function empty with \( \text{empty}(x) = \emptyset \) for each \( x \) is such that \( \tau \leq \text{empty} \) for each \( \tau \in \text{MSpec}(X) \).
3. If \((S, \leq)\) is finitary, and \( X \) is a finite set of variables, then for each \( \tau \in \text{MFIn0Spec}(X) \) the subposet of \( \text{Spec}(X) \) determined by the set \( \text{LBd}_{\text{Spec}}(\tau) = \{ p \in \text{Spec}(X) \mid p \leq \tau \} \) has a finite and minimal covering that we shall denote by \( \text{max}(\text{LBd}_{\text{Spec}}(\tau)) \).

**Proof**: To prove (1), just note that \( \text{LBd}(\cup_i (\tau_i(x))) = \bigcap_i \text{LBd}(\tau_i(x)) \). The proof of (2) follows easily from (1). Regarding (3), to find the desired minimal covering, denote by \( \text{max}(\text{LBd}(\tau(x))) \) the finite and minimal covering for \( \text{LBd}(\tau(x)) \) that exists by \( \tau(x) \) finite and nonempty; then \( \{ p \in \text{Spec}(X) \mid \forall x \in X, p(x) \in \text{max}(\text{LBd}(\tau(x))) \} \) is the desired family and is finite because \( X \) is finite. \( \square \)

In the following we will use a particularly simple type of finitary multiple sort assignment that we shall denote \( x \rightarrow s \). It is the function mapping \( x \) to \( \{s\} \) and mapping \( y \) to \( \emptyset \) when \( y \neq x \). Similarly, \( x \rightarrow s_1, \ldots, s_n \) will denote the multiple sort assignment mapping \( x \) to \( \{s_1, \ldots, s_n\} \) and all the other variables to \( \emptyset \). We shall say that a system of \( \Sigma \)-equations \((\Gamma, \rho_0)\) is nontrivial iff whenever \( x = y \) is in \( \Gamma \) with \( x \) and \( y \) variables, then \( \rho_0 \) is defined for either \( x \) or \( y \).

**Theorem 34**: For \((S, \leq, \Sigma)\) a coherent order-sorted signature the following are equivalent:

1. For each set of most general \( \Sigma \)-axioms \( E \) and system of nontrivial \( \Sigma \)-equations \((\Gamma, \rho_0)\) such that \( \text{Un}_{||\Sigma|E}(\Gamma) \) has a finite and minimal covering, \( \text{Un}_{\Sigma,E}(\Gamma, \rho_0) \) also has a finite and minimal covering.
(2) \((S, \leq, \Sigma)\) is finitary.

**Proof:** To see that \((1) \Rightarrow (2)\) assume that \((S, \leq, \Sigma)\) is not finitary, i.e., either (i) there is a nonempty finite \(S' \subseteq S\) such that \(LBd(S')\) does not have a finite and minimal covering or (ii) there are \(\sigma\) in \(\Sigma\), \(s\) in \(S \cup \{\ast\}\) and \(n\) in \(\mathbb{N}\) such that the set \(LBd(\sigma, s, n)\) has no finite and minimal covering. In case (i), consider the system of equations with \(\Gamma = \{x_s = x_{s'}, \ s', s'' \in S'\}\) and \(\rho_0\) the function assigning to each \(x_s\) the sort \(s\). The set of equations \(\Gamma\) has an initial unsorted (sober) unifier \(\theta\) mapping all the \(x_s\) to the same variable \(x\) and then the family \(\{(\theta, L_{E}(x)^{\sigma}_{\rho_0}) \mid \rho_0(x) = s, s \in LBd(S')\}\) is an initial family of unifiers for \((\Gamma, \rho_0)\) but does not contain a minimal finite initial family by our assumption on \(S'\). In case (ii), consider \(\Gamma = \{x = \sigma(y_1, \ldots, y_n)\}\) and \(\rho_0\) the partial function mapping \(x\) to \(s\) and otherwise undefined in case \(s \in S\), or the totally undefined function in case \(s = \ast\). The initial unsorted (sober) unifier \(\theta\) is the identity on the \(y_1, \ldots, y_n\) and maps \(x\) to \(\sigma(y_1, \ldots, y_n)\). The family

\[
\{(\theta, L_{E}(y_1, \ldots, y_n)^{\sigma}_{\rho_0}) \mid \rho_0(y_1, \ldots, y_n) = s, s \in LBd(\sigma, s, n)\}
\]

is an initial family of unifiers for \((\Gamma, \rho_0)\) but does not have a finite minimal initial family by our assumptions on \(LBd(s, \sigma, s, n)\).

Let \(\mathcal{M}\) be a finite minimal covering of unsorted \(\mathcal{L}E\)-unifiers of \(\Gamma\). The proof that \((2) \Rightarrow (1)\) will exhibit an algorithm to compute for each \(\theta: X \to T_{(\mathcal{E}, \mathcal{L}E)}(Y)\) in \(\mathcal{M}\) a finite set of finitary nonempty multiple sort assignments \(\mathcal{M}(\theta) = \{\tau_j\}\) of the variables \(Y\) such that \(\rho\) is such that \((\theta, L_{E}(Y)^{\rho}_{\rho_0})\in Un_{\mathcal{L}E}(\Gamma, \rho_0)\) iff there is some \(j\) such that \(\rho_j \leq \tau_j\). By Lemma 33 (3), for each \(\tau_j\) there is a finite set of sort assignments \(\{\rho_{j,i}\}_{i}\) that are maximal below \(\tau_j\), the family \(\{\rho_{j,i}\}_{i}\) is a finite covering of all sort assignments \(\rho\) such that \((\theta, L_{E}(Y)^{\rho}_{\rho_0})\in Un_{\mathcal{L}E}(\Gamma, \rho_0)\) and therefore contains a minimal subcovering. By gathering all such minimal subcoverings for all \(\theta\) in \(\mathcal{M}\), using Theorem 22 we get the desired finite family of order-sorted unifiers. The computation of the family \(\mathcal{M}(\theta)\) will use the algorithm \(IP_{\mathcal{E}}\) that we define below. \(IP_{\mathcal{E}}\) takes as input a finite set of pairs \((s, t)\) with \(t\) in \(T_{\mathcal{L}E}(Y)\) and \(s\) in \(S \cup \{\ast\}\) and \(t\) not a variable if \(s = \ast\), and returns a finite family of finitary sort assignments of the variables \(Y\). It is defined as follows:

- \(IP_{\mathcal{E}}(\{(t_i, s_i)\}_{i \in I}) = \bigcup_{i \in I} IP_{\mathcal{E}}(t_i, s_i)\).
- \(IP_{\mathcal{E}}(y, s) = \{(\ast, y)\}\) if \(s\) is max-arity(\(\sigma, s, 0\)) and \(\emptyset\) otherwise.
- \(IP_{\mathcal{E}}(\sigma(t_1, \ldots, t_n), s) = \bigcup_{i=1}^{\max-arity(\sigma, s, n)} IP_{\mathcal{E}}(t_i, s)^{\sigma}_{\rho_0}\).
- \(IP_{\mathcal{E}}(\sigma(t_1, \ldots, t_n), s) = \{\emptyset\}\) if \(\max-arity(\sigma, s, n) = 0\).

For each \(\theta: X \to T_{(\mathcal{E}, \mathcal{L}E)}(Y)\) in \(\mathcal{M}\) presented by \(\theta': X \to T_{\mathcal{L}E}(Y)\), we define \(\mathcal{M}(\theta)\) to be \(\mathcal{M}(\theta) = IP_{\mathcal{E}}(\{(t_i, s_i) \mid i \in I\})\) where \(I = \text{Dom}(\rho_0)\cap \Gamma_{\theta'}\) with \(\Gamma_{\theta'}\) the equations \(t = t'\) in \(\Gamma\) such that \(\theta'(t) \neq \theta(t)\) are not both variables and:

- For \(x \in \text{Dom}(\rho_0)\), \(t_x = \theta'(x)\) and \(s_x = \rho_0(x)\).
- For \((t_1 = t_2) \in \Gamma_{\theta'}, \ t_{(t_1 = t_2)} = \theta'(t_1)\) and \(s_{(t_1 = t_2)} = \ast\).

(Note that the equations \(E\) are most general and that \(\theta\) is a \(\mathcal{L}E\)-unifier, and therefore the choice of presentation and of \(t_1\) or \(t_2\) is immaterial). It is not hard to check that the finitary sort assignments \(\tau\) in \(\mathcal{M}(\theta) = IP_{\mathcal{E}}(\{(t_i, s_i) \mid i \in I\})\) are in fact nonempty, and to show that any sort
assignment $\rho$ of $Y$ leads to a unifier $(\theta, \sigma_{\leq}(Y^\rho))$ iff $\rho \preceq \tau$ for at least one such $\tau$. Checking that the definition does not depend on the choice of the presentation $\theta'$ is analogous to the argument in the characterization of normal signatures. \]

**Remark 35:** The IP algorithm first appeared in a draft of [Goguen & Meseguer 88b] distributed at the 1985 Bad-Honnef conference and in [Goguen, Jouannaud & Meseguer 85]; it has been further refined in [Kirchner, Kirchner & Meseguer 88].

**Corollary 36:** For $(S, \leq, \Sigma)$ a coherent order-sorted signature the following are equivalent:

1. For each system of nontrivial $\Sigma$-equations $(\Gamma, \rho_0)$, $\text{Un}_<^\Sigma(\Gamma, \rho_0)$ has a finite and minimal initial family.
2. $(S, \leq, \Sigma)$ is finitary.

\]

**Example 37:** Let $\Sigma$ be the finite (and therefore finitary) signature shown in Figure 1, and let $\Gamma = \{f(h(x)) = f(y)\}$, with $\rho_0$ defined by $\rho_0(x) = s_8$ and $\rho_0(y)$ is undefined.

Then the substitution $\theta: (x, y) \rightarrow T_{s_8}(x)$ leaving $x$ fixed and sending $y$ to $h(x)$ is the initial object of $\text{Un}_<^\Sigma(\Gamma)$. A minimal initial family for $\text{Un}_<^\Sigma(\Gamma, \rho_0)$ can be extracted from the set of all maximal sort assignments smaller than some $\tau$ in $\text{MS}(\theta)$. Using the IP algorithm we have

$\text{MS}(\theta) = \text{IP}((x, s_8), f(h(x)), \tau)) = \{ (x \rightarrow s_8) \cup \tau \}$

We also have

$\text{IP}(f(h(x)), \tau) = \text{IP}(f(h(x)), s_7) \cup \text{IP}(f(h(x)), s_2) \cup \text{IP}(f(h(x)), s_3) = $
IP(h(x),s_4) \cup IP(h(x),s_5) \cup IP(h(x),s_6) = \{(x \to s_4), (x \to s_5), (x \to s_6)\}.

Therefore, we have
\[ M(\emptyset) = \{(x \to s_4, s_4), (x \to s_5, s_5), (x \to s_6, s_6)\}. \]
The only two maximal sort assignments below some \( \tau \) in \( M(\emptyset) \) are \((x \to s_4)\) and \((x \to s_1, t)\). The minimal initial family is given by \((\emptyset, \tau(x^{(x \to s_4)})), (\emptyset, \tau(x^{(x \to s_1)}))\).

**Definition 38:** A partially ordered set \((S, \leq)\) is called **unitary** iff each nonempty finite subset \( S' \subseteq S \) such that the the poset \((LBD(S'), \leq)\) is nonempty has a glb. An order-sorted signature \((S, \leq, \Sigma)\) is called **unitary** iff

- \((S, \leq)\) is unitary, and
- for each \( \sigma \in \Sigma, s \in S \cup \{\ast\} \) and \( n \in \mathbb{N} \) if the set \( LBD(\sigma, s, n) \) is nonempty it has a most general element, i.e., \( \text{max-arity}(\sigma, s, n) \) is a singleton. \(^{11}\)

**Remark 39:** Notice that when a signature is unitary, since \( \text{max-arity}(\sigma, \ast, n) \) is either empty or a singleton, all operations \( \sigma \) have a maximum arity and coarity, so that all other instances of the operation are restrictions. In particular, no "ad hoc polymorphism" is permitted for unitary signatures, where "ad hoc polymorphism" means that we have \( \sigma: s_1 \ldots s_n \to s \) and \( s', 1 \ldots s', \to s' \) in \( \Sigma \) with \( s_i \) and \( s'_i \) in different connected components for some \( i \) or \( s \) and \( s' \) in different connected components, as when \( + \) is used for both integer addition and boolean exclusive or. By contrast, finitary or normal signatures can be ad hoc. Of course, it is a simple matter to modify the syntax of a given order-sorted signature so that ad hoc polymorphism is avoided while preserving previously existing overloading in each connected component.

**Lemma 40:** For \((S, \leq)\) a unitary poset, \( X \) a finite set of variables, and \( \tau \in \text{MFinn} \), the subposet of \( \text{Spec}(X) \) determined by the set \( LBD_{\text{Spec}}(\tau) = \{ \rho \in \text{Spec}(X) \mid \rho \preceq \tau \} \) has a minimal covering \( \text{max}(LBD_{\text{Spec}}(\tau)) \) that is either empty or consists of a single sort assignment.

**Theorem 41:** For \((S, \leq, \Sigma)\) a coherent order-sorted signature the following are equivalent:

1. For each set of most general \( \Sigma \)-axioms \( E \) and system of nontrivial \( \Sigma \)-equations \((\Gamma, \rho_0)\) such that \( \text{Un}_E(\Gamma) \) has a most general unifier, if \( \text{Un}_E(\Gamma, \rho_0) \) is nonempty, then it also has a most general unifier.

2. \((S, \leq, \Sigma)\) is unitary.

**Proof:** The proof \((1) \Rightarrow (2)\) is entirely similar to that for the finitary case. The proof of \((2) \Rightarrow (1)\) is just a specialization of the proof for the finitary case, noticing that now \( \text{IP}_x \) returns a family that is either empty, or consisting of a single finitary nonempty multiple sort assignment.

**Corollary 42:** **Order-Sorted Herbrand-Robinson Theorem** For \((S, \leq, \Sigma)\) a coherent order-sorted signature the following are equivalent:

1. For each system of \( \Sigma \)-equations \((\Gamma, \rho_0)\) if \( \text{Un}_E(\Gamma, \rho_0) \) is nonempty, then it has a most general unifier that is an initial object.

\(^{11}\)This condition on a signature \( \Sigma \), without the requirement that \((S, \leq)\) be unitary and excluding \( s=\ast \), is called **coregularity** in [Goguen & Meseguer 88b], because of its duality to the regularity condition.
2. $(S, \leq, \Sigma)$ is unitary.

8 A Quasi-Linear Order-Sorted Unification Algorithm

In practice, there is no need to separate the computation of the unsorted unifier from the type-checking process provided by the IP algorithm. In this section we give a variant of the Martelli-Montanari quasi-linear unification algorithm [Martelli & Montanari 82] that has the same complexity as its unsorted ancestor, and finds the initial unifier or stops with failure when presented with a system of $\Sigma$-equations $(\Gamma, \rho_0)$ for $\Sigma$ a finite unitary signature. In a sense, the order-sorted version is considerably better than the unsorted version, since failures can be caught much earlier thanks to the type-checking of the IP algorithm and the intersection of sorts that is performed. This yields additional support for the experience of [Walther 84] that order-sortedness can lead to drastic reductions of the search space. We begin with a quick summary of the unsorted Martelli-Montanari unification algorithm, and then explain how it is adapted to yield a quasilinear order-sorted unification algorithm. The style of exposition is informal, but we implicitly rely on the detailed proofs given in [Martelli & Montanari 82] to justify our claims.

8.1 The Unsorted Martelli and Montanari Algorithm

Unification is understood as the process of solving a system of multiequations. By a multiequation is meant an expression of the form $V = M$ where $V$ is a finite nonempty set of variables and $M$ is a finite multiset of terms. Given a finite multiset of terms $M$, the common part $C = C(M)$ is the biggest (i.e., the most instantiated possible) nonvariable term that has all terms in $M$ as instances and, in addition, for each variable $x$ occurring at position $\alpha$ in $C$ there is a term $t$ in $M$ such that $x$ occurs in the same position $\alpha$ in $t$ (the common part $C$ may not exist, in which case the terms in $M$ are not unifiable); the frontier $F = F(M)$ is a set of multiequations, one for each variable occurrence $\alpha$ in the common part $C$ of $M$, such that each multiequation in $F$ is made up of the subterms $t/\alpha$ at occurrence $\alpha$ for all $t$ in $M$ by grouping variables on the left and nonvariables on the right. A multiequation $V = M$ can be replaced by the set of multiequations $(V = C(M)) \cup F(M)$ which is equivalent to the original multiequation in the sense of admitting the same solutions. Such a replacement process is called multiequation reduction. We can illustrate these notions with a simple example. Consider the multiequation $V = M$ with $V = \{s\}$ and $M = \{f(g(x), h(y, g(z))), f(g(u), h(g(x), y)), f(g(y), h(v, w))\}; then $C(M) = f(g(x), h(y, y))$ and $F(M) =$

$$
\begin{align*}
\{x, y\} &= \{g(u)\} \\
\{y, v\} &= \{g(z)\} \\
\{y, w\} &= \{g(z)\}
\end{align*}
$$
Multiequation reduction applied to \( V = M \) then yields the following set of multiequations,

\[
\begin{align*}
\{ s \} &= \{ f(g(x), h(y, y)) \} \\
\{ x, y \} &= \{ g(u) \} \\
\{ y, v \} &= \{ g(z) \} \\
\{ y, w \} &= \{ g(z) \}
\end{align*}
\]

The process of **compactification** is one by which multiequations that have variables in common are merged together by taking the union of their lefthand and righthand sides, so that we obtain an equivalent new set of multiequations with disjoint lefthand sides. For example, our last set of multiequations compactifies to

\[
\begin{align*}
\{ s \} &= \{ f(g(x), h(y, y)) \} \\
\{ x, y, v, w \} &= \{ g(u), g(z) \}
\end{align*}
\]

The Martelli-Montanari algorithm operates on a pair \((T, U)\) where \(U\) is a set of multiequations and \(T\) is an (initially empty) sequence of multiequations such that:

- all sets of variables in lefthand sides of \(T\) and \(U\) are disjoint,
- in \(T\), all righthand sides consist of only one term,
- all variables in the lefthand side of some multiequation in \(T\) can only occur in the righthand side of a preceding multiequation in \(T\).

The pair \((T, U)\) is modified by the algorithm until either a failure is reached, or the algorithm stops with \(U\) empty and with a sequence \(T\) from which we can read off the desired unifier by associating to each multiequation \(\{x_1, ..., x_n\} = \{t\}\) in \(T\) its corresponding unifier (mapping the \(x_1, ..., x_n\) to \(t\)) and then composing them. Each multiequation has an associated **counter** that keeps track of the number of occurrences of its variables; we write \(V[n] = M\) for the multiequation \(V = M\) when its associated counter has value \(n\); the use and updating of counters for multiequation selection is further explained below. Now here is the algorithm:

(1) **repeat**

(1.1) select a multiequation in \(U\) of the form \(V[0] = M\) and otherwise, stop with failure; --(cycle)

(1.2) if \(M\) is empty **then** transfer the multiequation to the end of \(T\) **else**

begin

(1.2.1) compute the common part \(C\) and the frontier \(F\) of \(M\). If \(M\) has no common part, stop with failure; --(clash)

(1.2.2) transform \(U\) using multiequation reduction on the selected multiequation and compactification;

(1.2.3) transfer the multiequation \(V[0] = \{C\}\) from \(U\) to the end of \(T\);

end

until \(U\) is empty;

(2) **stop** with success.

A system of equations \(\Gamma = \{ t_i = t'_i \mid i = 1, ..., n \}\) is transformed into the pair

\[
(T = \emptyset, U = \{ \{ y_i \}[0] = \{ t_i, t'_i \} \mid i = 1, ..., n \} \cup \{ \{ x_j \}[n_j] = \emptyset \mid j = 1, ..., m \})
\]
where \(\{x_1, \ldots, x_m\}\) are the variables occurring in \(\Gamma\) and \(n_j\) is the total number of occurrences of \(x_j\) in \(\Gamma\). Therefore, each multiequation has an associated counter. Counters are updated in the following fashion: whenever an occurrence of a variable \(x\) appears in the lefthand side of a multiequation in a frontier \(F(M)\) computed during a multiequation reduction process, decrease by 1 the counter of the multiequation in \(U\) having the variable \(x\); whenever several multiequations are merged, the counter of the resulting multiequation is set equal to the sum of the counters of the multiequations being merged. The point of using counters is one of efficiency: it guarantees that the variables in the lefthand side of the multiequation chosen for reduction do not occur elsewhere in \(U\) and in this way saves a step of variable substitution that would otherwise be necessary.

Here is an example showing the different stages of the computation of the algorithm. It is the unsorted version of an order-sorted example presented later.

**Example 43:** Let \(\Gamma = \{f(h(x,x),p(g(z),r(q(z)))) = f(h(g(u),g(k(v,q(w)))),p(x,y))\}\). We then have the following steps of computation:

\[
T_0 = \emptyset
\]
\[
U_0 = \{
\{a\} \ [0] = \{f(h(x,x),p(g(z),r(q(z)))), f(h(g(u),g(k(v,q(w)))),p(x,y))\},
\{x\} \ [3] = \emptyset,
\{y\} \ [1] = \emptyset,
\{z\} \ [2] = \emptyset,
\{u\} \ [1] = \emptyset,
\{v\} \ [1] = \emptyset,
\{w\} \ [1] = \emptyset
\}
\]

\[
T_1 = \{a\} \ [0] = \{f(h(x,x),p(x,y))\}
\]
\[
U_1 = \{
\{x\} \ [0] = \{g(u), g(k(v,q(w))), g(z)\},
\{y\} \ [0] = \{r(q(z))\},
\{z\} \ [2] = \emptyset,
\{u\} \ [1] = \emptyset,
\{v\} \ [1] = \emptyset,
\{w\} \ [1] = \emptyset
\}
\]

\[
T_2 = \{a\} \ [0] = \{f(h(x,x),p(x,y))\},
\{x\} = \{g(u)\}
\]
Therefore, the most general unifier $\theta$ is given by $\theta(z) = \theta(u) = k(v, q(w)), \theta(x) = g(k(v, q(w))), \theta(y) = r(q(k(v, q(w))))$. $\square$
8.2 The Order-Sorted Algorithm

The order-sorted version of the Martelli-Montanari algorithm is quite similar, except that, each
multiequation has a sort associated to it, as well as a counter. Besides updating the counter of
a multiequation exactly as in the unsorted case, its sort must also be updated. We assume that
we have precomputed in a table the glb or intersection $s \& s'$ of any two sorts when it is defined,
or, by convention, the value $\emptyset$ when $s$ and $s'$ have no common lower bound; also by
convention, we introduce an additional sort, $\ast$, such that $s \& \ast = \ast \& s = s$ for any $s$. We also
assume that the value $\text{max-arity}(\sigma, s, n)$ is stored in another table. Therefore both the
intersection of two sorts, and the maximum arity of an operator, can be obtained in constant
time. We will use multiequations $V[n, s] = M$, where $n$ is a counter and $s$ is a sort (or the value
$\ast$). The system of equations $(T = \{t_i = t_i' \mid i = 1, \ldots, n\}, \rho_0)$ is transformed into the pair
$(T = \emptyset, U = \{y_1[0, \ast] = t_1, \ldots, t_n[0, \ast] \} \cup \{\{x_j[n_j, \rho_0(x_j)] = \emptyset \mid j = 1, \ldots, m\} \}$
where $\{x_1, \ldots, x_m\}$ are the variables occurring in $T$ and $n_j$ is the total number of occurrences of $x_j$ in $T$, and where
in general $T$ is a sequence of multiequations with all righthand sides containing only one term,
and with variables from lefthand sides only occurring in righthand sides of multiequations
earlier in the sequence, and where $U$ is a set of multiequations. Here is the algorithm; it either
stops with failure or with solution $T$ (with $U$ empty) such that all righthand sides of $T$ are in
fact sets:

(1) repeat
(1.1) select a multiequation in $U$ of the form $V[0, s] = M$ and otherwise, stop with
failure; --- (cycle)
(1.2) if $M$ is empty then transfer the multiequation to the end of $T$ else
begin
(1.2.1) compute the common part $C$ and the frontier $F$ of $M$. If $M$ has no
common part, stop with failure; --- (clash)
(1.2.2) transform $U$ using multiequation reduction on the selected
multiequation and compactification;
(1.2.3) transfer the multiequation $V[0, s] = \{C\}$ from $U$ to the end of $T$;
end
until $U$ is empty;
(2) stop with success.

Here the common part $C$ for the multiequation $V[0, s] = \{u_1, \ldots, u_n\}$ is computed just as in the
unsorted case, but $\text{IP}(C, s)$ is computed simultaneously yielding sorts $s_1, \ldots, s_k$ associated with
the frontier$^{12}$ $F = \{W_1[?, s_1] = N_1, \ldots, W_k[?, s_k] = N_k\}$ or failing.

As in the unsorted case, multiequation reduction transforms $U$ into $(U \setminus (V[0, s] = M)) \cup (V[0, s]$

---

$^{12}$ For multiequations in the frontier, the value of the counter is not defined, which by convention is denoted "$\?".
Counter updating was discussed in Section 8.1.
and compactification groups together multiequations whose left-hand sides have variables in common. However, now besides updating the counters as in the unsorted case, we must take the intersection of all the sorts in the multiequations being merged together, or fail in case that intersection does not exist. For example, in the first step of multiequation reduction in the example below (with signature in Figure 2) we obtain a common part \( C = f(h(x,x), p(x,y)) \) and a frontier

\[
F = \{
\{x\} [7, s_2] = [g(u)]
\{x\} [7, s_2] = [g(k(v, q(w)))]
\{x\} [7, s_3] = [g(z)]
\{y\} [7, s_3] = [r(q(z))] \}
\]

Compactification of the multiequations involving the variable \( x \) with the multiequation

\[
\{x\} [3, s_3] = \emptyset
\]

yields the multiequation

\[
\{x\} [0, s_3] = [g(u), g(k(v, q(w))), g(z)]
\]

As in the original Martelli-Montanari algorithm, the complexity is \( O(n \log(v)) \) for \( n \) the number of nodes in \( \Gamma \) and \( v \) the number of variables in \( \Gamma \). By a different implementation of the set operations one can obtain a complexity \( O(n \cdot G(V)) \) with \( G \) the inverse Ackerman's function and \( V \) the total number of variable occurrences in \( \Gamma \).

**Example 44:** Let \( \Sigma \) be the unitary signature shown in Figure 2, and let

\( \Gamma = \{f(h(x,x), p(g(z), r(q(z)))) = f(h(g(u), g(k(v, q(w)))), p(x,y))\} \) with \( \rho_0 \) sending \( x \) to \( s_5 \), \( v \) to \( s_7 \) and undefined elsewhere. We then have the following steps of computation:

\[
T_0 = \emptyset
\]

\[
U_0 = \{
\{a\} [0, *] = \{f(h(x,x), p(g(z), r(q(z))))\}
\{x\} [3, s_3] = \emptyset
\{y\} [1, *] = \emptyset
\{z\} [2, *] = \emptyset
\{u\} [1, *] = \emptyset
\{v\} [1, s_7] = \emptyset
\{w\} [1, *] = \emptyset
\}
\]

\[
T_1 = \{
\{a\} [0, *] = \{f(h(x,x), p(x,y))\}
\}
\]

\[
U_1 = \{
\{x\} [0, s_3] = [g(u), g(k(v, q(w))), g(z)]
\{y\} [0, s_3] = [r(q(z))]
\{z\} [2, *] = \emptyset
\{u\} [1, *] = \emptyset
\{v\} [1, s_7] = \emptyset
\{w\} [1, *] = \emptyset
\}
\]
Figure 2: Signature for Example 44

\[
T_2 = \\
\{a\}[*] = \{f(h(x,x),p(x,y))\} \\
\{x\}[s_8] = \{g(u)\} \\
U_2 = \{ \\
\{y\}[0,s_9] = \{r(q(z))\} \\
\{z,u\}[1,s_8] = \{k(v,q(w))\} \\
\{v\}[1,s_7] = \emptyset \\
\{w\}[1,*] = \emptyset \}
\]
Therefore, the most general and initial unifier is given by

\[ \theta: \langle x,y,z,u,v,w \rangle \rightarrow \tau E(v,w)^{\langle v \rightarrow s_{11}, v \rightarrow s_{11} \rangle} \]

with \( \theta \) leaving \( v \) and \( w \) fixed and with \( \theta(z)=\theta(u)=k(v,q(w)), \theta(x)=g(k(v,q(w))), \theta(y)=r(q(k(v,q(w)))) \).

Acknowledgements. We are grateful to all the members of the OBJ group with whom we have developed ideas and examples of order-sorted algebra and specially to Prof. Jean-Pierre
Jouannaud and Drs. Claude and Hélène Kirchner with whom the first two authors have developed the related theory of order-sorted term rewriting. Special thanks to Prof. Ugo Montanari who gladly oriented our attempt at making his algorithm order-sorted. We also thank Mr. Tomás Isakowitz, Dr. Mark Stickel and the three anonymous referees for their very valuable comments.

References


Order-Sorted Unification

[Goguen & Meseguer 86]

[Goguen & Meseguer 87a]

[Goguen & Meseguer 87b]

[Goguen & Meseguer 88a]

[Goguen & Meseguer 88b]

[Goguen, Jouannaud & Meseguer 85]

[Harper, MacQueen & Milner 86]


[Huet & Hullot 82]
[Kirchner, Kirchner & Meseguer 88]

[Knuth & Bendix 70]


[Meseguer & Goguen 85]


[Smolka, Nutt, Goguen & Meseguer 88]


