January 1988

Order-Sorted Congruence Closure

Jean H. Gallier  
*University of Pennsylvania, jean@cis.upenn.edu*

Tomas Isakowitz  
*University of Pennsylvania*

Follow this and additional works at: [http://repository.upenn.edu/cis_reports](http://repository.upenn.edu/cis_reports)

**Recommended Citation**  


This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/cis_reports/686](http://repository.upenn.edu/cis_reports/686)  
For more information, please contact repository@pobox.upenn.edu.
Order-Sorted Congruence Closure

Abstract
In this paper, an algorithm for testing the unsatisfiability of a set of ground order-sorted equational Horn clauses (for coherent signatures) is presented. This result follows from the fact that the concept of congruence closure extends to finite sets of ground order-sorted equational Horn clauses. We show how to compute the order-sorted congruence closure and obtain an algorithm running in $O(\eta^2)$.

Comments
ORDER-SORTED
CONGRUENCE CLOSURE

Tomas Isakowitz
Jean H. Gallier

MS-CIS-88-07
LINC LAB 96

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

January 1988

Acknowledgements: This research was supported in part by DARPA grant N00014-85-K-0018, NSF grants DCR-86-07156 and DCS-8219196-CER and U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027.
Abstract

In this paper, an algorithm for testing the unsatisfiability of a set of ground order-sorted equational Horn clauses (for coherent signatures) is presented. This result follows from the fact that the concept of congruence closure extends to finite sets of ground order-sorted equational Horn clauses. We show how to compute the order-sorted congruence closure and obtain an algorithm running in $O(n^2)$. 
## Contents

1 Introduction ................................. 1

2 Order-Sorted Algebras ..................... 1
   2.1 Signatures .................................. 1
   2.2 Algebras .................................... 3
   2.3 Order-sorted term algebra ............... 4
   2.4 Equations ................................... 5
   2.5 Order-sorted Logic ......................... 5
   2.6 Transforming Horn clauses into equational form ............. 7
   2.7 Order-sorted relations ................. 8
   2.8 Order-sorted congruences ............... 9
   2.9 The top algebra ............................ 10
   2.10 Least Order-sorted congruences ....... 13

3 Congruences Associated With Order-sorted Horn Clauses ...... 16
   3.1 The Graph $GT(H)$ .......................... 16
   3.2 Order-sorted Graph Congruence Closure ............... 17
   3.3 A Method For Testing Unsatisfiability .............. 18
   3.4 Finite models ................................ 22

4 Existence of the Order-sorted Graph Congruence Closure ...... 23
   4.1 Equational Order-sorted Graph Congruence Closure ....... 23
   4.2 Implicational Closure ....................... 25
   4.3 Order-sorted Congruence Closure For Horn Clauses ....... 25
   4.4 A more efficient Order-sorted Graph Congruence closure 26

5 Conclusion .................................. 27
1 Introduction

Order-sorted algebras were first introduced by Goguen and Meseguer [6], in order to deal with the notion of subtype and the notion of error in abstract data types. They constitute an interesting extension of many-sorted algebras, and their properties have been investigated recently by Goguen and Meseguer [5], Nutt, Smolka, Goguen, and Meseguer [14], and by Ait-Kaci and Smolka [15], where they are used to provide a semantics for inheritance.

In spite of this recent interest, some basic logical properties of these algebras have been neglected, in particular, the notion of congruence closure [8,9,13]. In this paper, it is shown that the notion of congruence closure extends to finite sets of ground order-sorted equations. Actually, extending a result of Gallier [2], it is shown that congruence closure extends to finite sets of ground order-sorted equational Horn clauses. More specifically, we prove that congruence closure is sound and complete for showing the unsatisfiability of sets of ground order-sorted equational Horn clauses (for coherent signatures). As a consequence, adapting results from Gallier [2], an unsatisfiability algorithm running in $O(n^2)$ is obtained.

Due to the nature of subtypes in order-sorted algebras, the proof that the method is complete is surprisingly nontrivial. This is because the approach used in the many-sorted case does not work. An approach using rewriting (as in Kozen [9]) fails, due to the well known problem that rewriting with order-sorted rewrite rules may create ill-typed terms. Also, the construction of a finite counter example as in Gallier [2] does not quite work, because it does not seem possible to define directly a finite algebra and respect the type structure at the same time. Our solution is to define a counter example by taking the quotient of the initial order-sorted algebra by the least congruence $\simeq$ containing the graph congruence closure $\rightarrow^E$ of the set of clauses. But then, we run into the problem that it is not obvious that congruence modulo $\simeq$ is conservative over congruence modulo $\leftrightarrow^E$ for terms occurring in the clauses. However, we are able to prove this result by characterizing congruence modulo $\simeq$ in a way that imitates congruence closure. Thus, we are able to prove the completeness (and soundness) of this congruence closure method (for ground Horn clauses).

2 Order-Sorted Algebras

2.1 Signatures

Our definition of an order-sorted algebra is equivalent to that given by Goguen and Meseguer [6] and by Kirchner and Kirchner [7], but we believe that it is slightly easier to grasp.

Given an index set $S$, an $S$-sorted set $A$ is just a family $(A_s)_{s \in S}$ of sets, one set $A_s$ for each $s \in S$. Similarly, given two $S$-sorted sets $A$ and $B$, an $S$-sorted function $f : A \rightarrow B$ is an $S$-indexed family $(f_s : A_s \rightarrow B_s)_{s \in S}$ of functions $f_s : A_s \rightarrow B_s$, and an $S$-sorted relation $R$ is an $S$-indexed family $(R_s)_{s \in S}$ of relations $R_s \subseteq A_s \times B_s$. Let us assume a fixed set $S$ called the sort set, with a partial order $\leq$. 

**Definition 2.1** A many-sorted signature is defined as a triple \((S, \Sigma, \rho)\), where \(S\) is a sort set and \(\rho : \Sigma \to \mathbb{S}^S\times S\) is a rank function assigning a set \(\rho(f)\) of ranks \((w, s)\) to each symbol in \(\Sigma\). The elements of the sets \(\Sigma\) are called operators or function symbols. The set \(\Sigma\) can be viewed as an indexed family if for every \((w, s) \in S^* \times S\) we let \(\Sigma_{w,s} = \{f \in \Sigma \mid (w, s) \in \rho(f)\}\).

Note that \(\Sigma_{w,s}\) and \(\Sigma_{w',s'}\) are not necessarily disjoint, since a symbol in \(\Sigma\) may have several ranks. Whenever convenient, we omit the function \(\rho\), and view \(\Sigma\) as family of sets \((\Sigma_{w,s})_{(w,s) \in S^* \times S}\).

**Definition 2.2** An order-sorted signature is a quadruple \((S, \leq, \Sigma, \rho)\), such that \((S, \Sigma, \rho)\) is a many-sorted signature and \((S, \leq)\) is a partially ordered set.

When the sort set \(S\) is clear, we write \((\Sigma, \rho)\) or \(\Sigma\) for \((S, \Sigma, \rho)\). Similarly when the partially ordered set is clear, we write \((\Sigma, \rho)\) or \(\Sigma\) for \((S, \leq, \Sigma, \rho)\).

For function symbols, we may write \(f : w \mapsto s\) when \((w, s) \in \rho(f)\) to emphasize that \(f\) denotes a function with arity \(w\) and co-arity \(s\). An important case occurs when \(w = \lambda\), the empty string; then \(f\) denotes a constant of sort \(s\). When \((w, s) \in \rho(f)\) we will also say that \(f\) has arity \(w\) and co-arity \(s\).

**Example 2.1** Let the set of sorts be \(S = \{\text{zero, rat}^+, \text{rat}\}\), and let the partial order be: \(\text{zero} \leq \text{rat}, \text{rat}^+ \leq \text{rat}\)

The following is an order-sorted \(\Sigma\)-signature:

\[
\Sigma_{\text{rat}.\text{rat}} = \{+\}
\]

\[
\Sigma_{\text{rat}.\text{rat}^+.\text{rat}} = \{/\}
\]

Notice that the second argument of \(/\) is of sort \(\text{rat}^+\), which is intended to exclude \(\text{zero}\). Hence we trying to formalize the idea of disallowing a division by \(\text{zero}\).

In order that the standard construction of an initial algebra as a term algebra holds, we restrict ourselves to a special class of signatures called regular. Essentially, regularity asserts that overloaded operations are consistent under restrictions to subsorts. Note that the ordering \(\leq\) on \(S\) extends to an ordering on strings of equal length in \(S^*\) as follows: \(s_1 \cdots s_n \leq s'_1 \cdots s'_n\) iff \(s_i \leq s'_i\) for \(1 \leq i \leq n\). Similarly, \(\leq\) extends to pairs in \(S^* \times S\) by stating that \((w, s) \leq (w', s')\) iff \(w \leq w'\) and \(s \leq s'\).

**Definition 2.3** An order-sorted signature \(S\) is regular iff for every \(f \in \Sigma\), every \(w^0 \in S^*\), and every \((w, s) \in \rho(f)\), if \(w^0 \leq w\), then the set \(\{(w', s') \in \rho(f) \mid w^0 \leq w'\}\) has a least element.

When the set of sorts is finite (or well founded), regularity is captured by a combinatorial condition (see the paper by Kirchner and Kirchner [7]).
Lemma 2.1 An order-sorted signature $\Sigma$ over a finite (or well founded) sort set $S$ is regular iff for every every $f \in \Sigma$, every $w^0 \in S^*$, and every pair of ranks $(w, s), (w', s') \in \rho(f)$, if $w^0 \leq w, w'$, then the set $\{(w, s), (w', s')\}$ has a lower bound $(w_i, s_i)$ such that $(w_i, s_i) \in \rho(f)$, and $w^0 \leq w_i$.

Let $\equiv = (\leq \cup \leq^{-1})^+$ be the equivalence relation induced by the partial order $\leq$. We will say that to sorts $s$ and $s'$ are connected if $s \equiv s'$.

Notice that being connected is an equivalence relation that splits the set of sorts into connected components. The concept of connected sorts is important for defining quotient algebras. Indeed, in order for the usual construction of the quotient of an algebra by a congruence to hold, we need a condition on signatures called coherence.

**Definition 2.4** A regular order-sorted signature is coherent if every connected component has a greatest element called the top sort of the connected component.

### 2.2 Algebras

For any string $w = s_1, \ldots, s_n$ ($n \geq 1$), let $A_w = A_{s_1} \times \cdots \times A_{s_n}$, with $A_\lambda = \{\lambda\}$ (a one element set).

**Definition 2.5** Let $(S, \leq, \Sigma, \rho)$ be an order-sorted signature. An order sorted $(S, \leq, \Sigma, \rho)$-algebra $A$ is a pair $(A, I)$ consisting of an $S$-sorted family $A = (A_s)_{s \in S}$ called the carrier of $A$, and a function $I$ called the interpretation function of $A$, where $I$ assigns to every $f \in \Sigma$ an indexed family of functions $I(f) = (I^w_f : A_w \to A_s)_{(w, s) \in \rho(f)}$. In particular, when $w = \lambda$, $I^w_f \in A_s$ is an element of $A_s$. For each sort $s$, $A_s$ is the carrier of sort $s$. Note that the carrier of sort $s$ may be empty. We also denote $I^w_f$ as $f^{w,s}_A$. Moreover, the following conditions hold:

1. $A_s \subseteq A_{s'}$ whenever $s \leq s'$, and
2. If $(w, s) \in \rho(f)$ and $(w', s') \in \rho(f)$, $s \leq s'$, and $w \leq w'$, then $f^{w,s}_A : A_w \to A_s$ is equal to the restriction of $f^{w',s'}_A : A_{w'} \to A_{s'}$ to $A_w$. That is, for any $\overline{a} \in A_w$, $f^{w',s'}_A(\overline{a}) = f^{w,s}_A(\overline{a})$.

By abuse of notation, we may denote an algebra and its carrier by the same name unless confusions arise. For example in the the previous definition we might use $A$ for both the carrier (which is $A$) and for the algebra (which is $A$). We may also drop some of the components in $(S, \leq, \Sigma, \rho)$ when talking about order-sorted algebras, or drop the superscript $(w, s)$ when referring to a function $f^{w,s}_A$.

**Example 2.2** Consider the signature presented in example 2.1, an order-sorted $\Sigma$-algebra $A$ is:

- $A_{\text{rat}} = Q$ (the set of rational numbers),
\[ A_{\text{rat}}^+ = \mathbb{Q} - \{0\} \text{ (the set of non-zero rationals)}, \] 
\[ A_{\text{zero}} = \{0\}. \]

The functions have their natural interpretations:

- \( +_A \) is addition of rational numbers,
- \( /_A \) is division of rational numbers.

For any \( w = s_1 \ldots s_n \neq \lambda \) and \( \bar{a} = (a_1, \ldots, a_n) \in A_w \), let \( h_w(\bar{a}) = (h_{s_1}(a_1), \ldots, h_{s_n}(a_n)) \).

**Definition 2.6** Let \((S, \preceq, \Sigma, \rho)\) be an order-sorted signature, and let \( A \) and \( B \) be \((S, \preceq, \Sigma, \rho)\)-order-sorted algebras. A \((S, \preceq, \Sigma, \rho)\)-homomorphism \( h : A \mapsto B \) is an \( S \)-sorted function such that

1. for every constant \( c \) of sort \( s \), \( h_s(c_A) = c_B \),
2. for every \( f \in \Sigma \), every \((w, s) \in \rho(f)\), and every \( \bar{a} \in A_w \),
   \[ h_s(f^w_s(\bar{a})) = f^w_s(h_w(\bar{a})) \],
3. \( w \preceq w' \) and \( \bar{a} \in A_w \) implies \( h_w(\bar{a}) = h_w(\bar{a}) \).

When the partially ordered set is clear, \((S, \preceq, \Sigma, \rho)\)-homomorphisms are called order-sorted \( \Sigma \)-homomorphisms. We may also drop some of the components in \((S, \preceq, \Sigma, \rho)\) when talking about order-sorted homomorphisms.

### 2.3 Order-sorted term algebra

Following [6], we now define the order-sorted \( \Sigma \)-term algebra \( T_\Sigma \) as the least family \( \{T_s \mid s \in S\} \) of sets satisfying the following conditions:

1. \( \Sigma_{\lambda, s} \subseteq T_{\Sigma, s} \) for \( s \in S \);
2. \( T_{\Sigma, s} \subseteq T_{\Sigma, s'} \) whenever \( s \preceq s' \);
3. if \( f \in \Sigma_{w, s} \), and if \( t_i \in T_{\Sigma, s_i} \) where \( w = s_1, \ldots, s_i \neq \lambda \), then the string \( ft_1 \ldots t_n \) is in \( T_{\Sigma, s} \).

In addition, the function symbols are interpreted as string constructors as follows: for \( f \in \Sigma_{w, s} \), \( f^w_s(t_1, \ldots, t_n) = ft_1 \ldots t_n \).

**Definition 2.7** Let \( \Sigma \) be an order-sorted signature. An order-sorted algebra \( A \) is initial in the class of all \( \Sigma \)-algebras if there is a unique order-sorted \( \Sigma \)-homomorphism from \( A \) to any other \( \Sigma \)-algebra.

We state the following theorems, whose proofs can be found in [6].
Theorem 2.2 Let \( \Sigma \) be a regular order-sorted signature. Then \( T_\Sigma \) is an initial order-sorted \( \Sigma \)-algebra. This means that for every order-sorted \( \Sigma \)-algebra \( A \), there is a unique order-sorted homomorphism \( h_A : T_\Sigma \to A \).

Theorem 2.3 Let \( \Sigma \) be a regular order-sorted signature. Then every term \( t \) in \( T_\Sigma \) has a least sort denoted by \( LS(t) \).

2.4 Equations

An equation can be stated between two terms if they belong to the same top sort. Viewed as terms in \( T_\Sigma \) we can say that two terms can be equated if their least sorts are in the same connected component, because then they both belong to the same top sort \( t \) and the equality symbol is to be seen as that of sort \( t.t \).

Definition 2.8 Given a coherent order-sorted signature \( \Sigma \), an equation or equational atom is a pair \( (u, v) \) of terms in \( T_\Sigma \) such that the least sorts \( LS(u) \) and \( LS(v) \) of \( u \) and \( v \) are connected. An equation is also denoted as \( u \equiv v \).

The concept of validity of an equation is defined using the initiality property of \( T_\Sigma \).

Definition 2.9 Given a coherent order-sorted signature \( \Sigma \), we say that an equation \( u \equiv v \) is valid in some order-sorted \( \Sigma \)-algebra \( A \) (denoted \( \models A \models u \equiv v \)) if and only if

\[
h_{A,LS(u)}(u) = h_{A,LS(v)}(v),
\]

where \( h_A : T_\Sigma \to A \) is the unique order-sorted homomorphism given by initiality. We say that \( u \equiv v \) is universally valid, or valid, and (denoted \( \models u \equiv v \)) iff \( u \equiv v \) is valid in every order-sorted \( \Sigma \)-algebra \( A \).

2.5 Order-sorted Logic

A logic language consists of terms and predicates. We extend the notion of an order-sorted signature to include predicates.

Definition 2.10 Given a coherent order-sorted signature \( (S, \leq, \Sigma, \rho_\Sigma) \), an order-sorted signature with predicates is a quintuple \( (S, \leq, \Sigma, \Pi, \rho) \), where \( \Pi \) is a family of predicate symbols. The function \( \rho \) assigns ranks to the elements of \( \Sigma \) and \( \Pi \) so that

1. for \( f \in \Sigma, \rho(f) = \rho_\Sigma(f) \), and
2. for \( P \in \Pi, \rho(P) \in 2^S \), i.e. \( P \) might have more than one rank (even different number of arguments).
We also demand that the following regularity condition be satisfied:

For each $P \in \Pi$, every $w^0 \in S^*$, and every $w \in \rho(P)$, if $w^0 \leq w$, then the set
\[ \{w' \in \rho(P) \mid w^0 \leq w'\} \]
has a least element.

Notice that an order-sorted signature with predicates is by definition coherent. As in the case of order-sorted signature, we can regard $\Pi$ as an $S^*$-indexed family $\{\Pi_s\}_{s \in S^*}$. When the sort set $S$, the partial order $\leq$ and the rank function $p$ are clear, we write $(\Sigma, \Pi)$ for $(S, \leq, \Sigma, \Pi, \rho)$. A predicate with empty rank, i.e. $X \in \rho(P)$, is a propositional letter.

Notice that if $(S, \leq, \Sigma, \Pi, \rho)$ is an order-sorted signature with predicates, $(S, \leq, \Sigma, \rho \upharpoonright \Sigma)$ is an order-sorted signature where $\rho \upharpoonright \Sigma$ is the restriction of $\rho$ to $\Sigma$. The notion of a structure is obtained by adding interpretations to the predicate symbols in such a way that the subsort relation is preserved. Let $BOOL$ be the set \{true, false\}.

**Definition 2.11** Let $(S, \leq, \Sigma, \Pi, \rho)$ be an order-sorted signature with predicates. Then an $(S, \leq, \Sigma, \Pi, \rho)$-structure is a $(S, \leq, \Sigma, \rho \upharpoonright \Sigma)$-algebra $M$ together with an interpretation $P_{s_1 \ldots s_n}^M$ for each predicate $P \in \Pi$ and each $s_1 \ldots s_n \in \rho(P)$ such that

1. $P_{s_1 \ldots s_n}^M : M_{s_1 \ldots s_n} \mapsto BOOL$, and
2. whenever $w_1, w_2 \in \rho(P)$ with $w_1 \leq w_2$ and $x \in M_{w_1}$, $P_{s_1 \ldots s_n}^{w_1}(x) = P_{s_1 \ldots s_n}^{w_2}(x)$.

The second condition states that the interpretation is consistent under subsorts. Notice that for a propositional letter $A$, $A_M$ is a constant, either true or false.

Atoms are expressions of the form $P_{t_1 \ldots t_n}$ such that $P$ is a predicate symbol and $t_1, \ldots, t_n$ are $\Sigma$-terms of sort $s_1, \ldots, s_n$ for $s_1 \ldots s_n \in \rho(P)$. The notion of satisfaction is as in the many-sorted case:

**Definition 2.12** Let $\Sigma, \Pi$ be an order-sorted signature with predicates and let $M$ be a $\Sigma, \Pi$-structure. Given a $\Sigma, \Pi$-atom $P_{t_1 \ldots t_n}$ with $t_i$ of sort $w_i$ and $w = w_1 \ldots w_n \in \rho(P)$, we say that $M$ satisfies $P_{t_1 \ldots t_n}$ (written $M \models P_{t_1 \ldots t_n}$) if $P_{s_1 \ldots s_n}^w(t_1, \ldots, t_n) = true$.

We now introduce additional notation for Horn clauses. Let $\Sigma, \Pi$ be an order-sorted signature with predicates and equality.

**Definition 2.13** A $\Sigma, \Pi$-Horn clause is an expression of the form

$A : -B_1, \ldots, B_n$

where each of $A, B_1, \ldots, B_n$ is either an atom or an equational atom. $A$ is called the head and $B_1, \ldots, B_n$ the body. When the body is empty, the Horn clause is called a unit clause and written $A$ instead of $A : -$. If the head of a clause is empty, it is a negative clause.

The notion of satisfaction of ground Horn clauses is the usual one:
Transforming Horn clauses into equational form

Definition 2.14 Given a ground $\Sigma, \Pi$-Horn clause $A : -B_1, \ldots, B_n$, $M$ satisfies it if $M \models B_i$ for $i = 1, \ldots, n$ implies $M \models A$.

For $H$ a set of ground Horn clauses, we say that $M$ is a model of $H$, and we write $M \models H$ if $M$ satisfies every Horn clause in $H$.

Notice that $M$ satisfies a negative clause $-B_1, \ldots, B_n$ if and only if $M \not\models B_i$ for $i = 1 \ldots n$.

The syntax and semantics for the logical connectives $\neg, \lor, \land, \exists$ and $\equiv$ are as in the unsorted case. Since there are no variables in our discussion, all terms are ground. That is the reason no assignment is needed in defining satisfaction.

2.6 Transforming Horn clauses into equational form

For the purpose of our discussion later on, it is useful to regard arbitrary Horn clauses as equational ones. This is done by replacing each occurrence of a non-equational atom $P_{t_1} \ldots t_n$ by the equation $P_{t_1} \ldots t_n = T$. If we add to our language a special sort $bool$, a constant $T$ interpreted as $true$, and for every structure, the domain $BOOL$ of sort $bool$ is the set of truth values $\{true, false\}$, every atomic formula $P_{t_1} \ldots t_k$ is logically equivalent to $P_{t_1} \ldots t_k \equiv T$, in the sense that $P_{t_1} \ldots t_k \equiv (P_{t_1} \ldots t_k \equiv T)$ is valid. But then, this means that $\equiv$ behaves semantically exactly as the identity relation on $BOOL$. Hence, we can treat $\equiv$ as the equality symbol $=$ of sort $bool$.

For every predicate letter $P$ of rank $\rho(P) = \{s_{11}, s_{21}, \ldots, s_{1n}, s_{2n}, \ldots, s_{kn}^1, \ldots, s_{kn}^k\}$ a new function symbol $P$ of sort $\{(s_{11}, s_{21}, bool), \ldots, (s_{kn}^1, s_{kn}^k, bool)\}$ is introduced. Hence, every set $H$ of order-sorted Horn clauses is equivalent to a set $H'$ of order-sorted Horn clauses, in which every atomic formula $P_{t_1} \ldots t_k$ is replaced by the equation $P_{t_1} \ldots t_k = T$. Formally we proceed as follows.

Definition 2.15 An order-sorted signature with boolean symbols is an order-sorted signature where the set of sorts $S$ contains the special sort $bool$, which can only be the target type (or co-arity) of a function symbol. In addition we require the constants $T$ and $\bot$ of rank $bool$ to be in the signature. (There might also exist other function symbols of co-arity $bool$).

The semantics of these signatures is given by order-sorted algebras that interpret the objects related to $bool$ in a specific way.

Definition 2.16 An order-sorted-$\Sigma$-bool algebra is an order-sorted-$\Sigma$ algebra $A$ such that:

- $A_{bool} = \{true, false\}$,
- $\top_A = true$, and
- $\bot_A = false$. 

One can define equations, equational Horn clauses and the notion of satisfaction as in the case of normal order-sorted signature, except that the models now have a fixed interpretation for the sort \textit{bool} and its related function symbols.

Given an order-sorted signature with predicates \((S, \leq, \Sigma, \Pi, \rho)\), we transform it into an order-sorted signature with boolean symbols (but without predicates) \((S, \leq, \Sigma \cup \Pi, \rho^\sigma)\) by letting \(\rho^\sigma\) be:

- for \(f \in \Sigma\), \(\rho^\sigma(f) = \rho(f)\), and
- for \(P \in \Pi\), \(\rho^\sigma(P) = (\rho(P), \text{bool})\).

Let \(H\) be a set of Horn clauses over \(T(S, \leq, \Sigma, \Pi, \rho)\), we transform it into a set \(H'\) of equational Horn clauses over \(T(S, \leq, \Sigma \cup \Pi, \rho^\sigma)\) by replacing every occurrence of an atomic formula \(P t_1 \ldots t_n\) in \(H\) by the equation \(P t_1 \ldots t_n = \top\) in \(H'\). Then the following lemma holds:

**Lemma 2.4** \(H\) is satisfiable if and only if \(H'\) is satisfiable.

### 2.7 Order-sorted relations

**Definition 2.17** We say that an \(S\)-sorted relation \(R\) on an order-sorted algebra is order-sorted iff given elements \(u\) and \(v\) of sort \(s\), and given a sort \(t\) such that \(s \leq t\),

\[
    uR_s v \quad \text{iff} \quad uR_t v.
\]

Informally this means that the relation is independent of the specific sort within a connected component.

Not every relation is order-sorted but it can be made into one by simply adding the missing pairs.

**Definition 2.18** Given a relation \(R\), the \textbf{order-sorted completion} of \(R\) is the family \(\{R'_s\}_{s \in S}\):

\[
    R'_s = \{(u, v) \in A_s \times A_s \mid (u, v) \in R_t \text{ for some } t \text{ such that } t \geq s \text{ or } s \geq t\}
\]

We can ask ourselves the following question. Given an order-sorted algebra and an order-sorted relation \(R\), can it occur that \(uR_s v\) for some sort \(s\) but \(\neg uR_{s'} v\) for a different sort \(s'\)? It turns out that for initial algebras of a coherent signature this is impossible.

**Lemma 2.5** Let \(\Sigma\) be a coherent signature. Consider the term algebra \(T_\Sigma\). A relation \(\sim\) on \(T_\Sigma\) is order-sorted if and only if for every pair of terms \(u\) and \(v\) and every pair of sort \(\tau, \sigma\) such that both \(u\) and \(v\) are of sort \(\tau\) and \(\sigma\),

\[
    u \sim_{\tau} v \quad \text{iff} \quad u \sim_{\sigma} v
\]

**Proof:** Since \(\Sigma\) is regular and \(u\) is of sort \(\sigma\) and \(\tau\), \(\sigma\) and \(\tau\) are connected via the least sort of \(u\). Since \(\Sigma\) is coherent, there is a top element \(\delta\) in the connected component of \(\tau\) and \(\sigma\), i.e. \(\tau \leq \delta\) and \(\sigma \leq \delta\). If \(\sim\) is order-sorted, \(u \sim_{\tau} v\) iff \(u \sim_{\delta} v\) iff \(u \sim_{\sigma} v\).

The converse direction is trivial. \(\square\)
2.8 Order-sorted congruences

Since every initial algebra is isomorphic to a term algebra, we have as a corollary that any initial algebra satisfies the above property. Thus when dealing with order-sorted relations over an initial algebra of a coherent signature, we can drop the subscripts from the relation symbol without causing any confusion. That is, \( uR_s v \) can be written as \( uRv \), since we are guaranteed that whenever this last expression makes sense, it holds.

2.8 Order-sorted congruences

Congruences are equivalence relations which are preserved under function application. They are useful because just as one can define a quotient of a set by an equivalence relation, it is possible to define the quotient of an algebra by a congruence.

Definition 2.19 For \((S, \Sigma)\) a many-sorted signature and \(A\) a \(\Sigma\)-algebra, a relation \(\sim\) is a many-sorted congruence if the following conditions are satisfied:

1. \(\sim\) is an equivalence relation
2. For every \(f \in \Sigma_{s_1, \ldots, s_n, s}\), if \(u_i \sim_{s_i} v_i\) for every \(i, 1 \leq i \leq n\), then \(f_A(u_1, \ldots, u_n) \sim_s f_A(v_1, \ldots, v_n)\).

Definition 2.20 For \((S, \leq, \Sigma)\) an order-sorted signature and \(A\) an order-sorted \(\Sigma\)-algebra, a many-sorted congruence \(\sim\) is an order-sorted congruence if it is an order-sorted relation. That is, if it satisfies:

for every \(s, s' \in S\) and \(a, b \in A\), if \(s \leq s'\) then \(a \sim_s b\) iff \(a \sim_{s'} b\).

We proceed to define the quotient algebra.

Definition 2.21 For \((S, \leq, \Sigma)\) a coherent order-sorted signature, \(A\) an order-sorted \(\Sigma\)-algebra, and \(\sim\) an order-sorted \(\Sigma\)-congruence, the quotient of \(A\) by \(\sim\) is the order-sorted \(\Sigma\)-algebra \(A/\sim\) defined as follows:

- for each top sort \(t\), the carrier \((A/\sim)_t\) is \(A_t/\sim\),
- for each other sort \(s\) whose connected component has top sort \(\max(s)\), the carrier \((A/\sim)_s\) is \(q_{\max(s)}(A_s)\), where \(q_{\max(s)} : A_{\max(s)} \mapsto (A/\sim)_{\max(s)}\) is the natural projection \(a \mapsto [a]\) of each element \(a\) to its \(\sim_{\max(s)}\) equivalence class.
- The interpretation of each function symbol \(f\) of rank \((\sigma, s)\) is given by:

\[
[f^\sigma_{(A/\sim)}([a_1], \ldots, [a_n])] = [f^\sigma_{A}([a'_1], \ldots, [a'_n])]
\]

where \(\sigma = \sigma_1, \ldots, \sigma_n\) and \(a'_i \in [a_i] \cap A_{\sigma_i}\). The interpretation is well defined since \(\sim\) is an order-sorted \(\Sigma\)-congruence.
Notice that the carrier for a sort \( s \) is not simply \( A_s/\sim \), but the restriction of the quotient map on the top sort of \( s \). This is to ensure that \( (A/\sim)_s \subseteq (A/\sim)_{s'} \) whenever \( s \leq s' \).

The elements \( a'_i \), in 1 above, are needed because not every member of \([a_i]\) is of sort \( \sigma_i \), hence some of them might not belong to the domain of \( f_A \). As an example consider the following case.

**Example 2.3** Let the signature be such that the sort structure is \( S = \{s_1, s_2, s_3, s\} \) with \( s_1, s_2, s_3 \leq s \), \( \Sigma = \{a, b, c, f\} \), \( a, b \) and \( c \) are constants of rank \( s_1, s_2 \) and \( s_3 \) respectively, and \( f \) has rank \( \{(s_1, s_1), (s_3, s_3)\} \). The situation is depicted in figure 1.

Consider the initial algebra \( T_\Sigma \). Notice that \( s_2 \) is not in the arity of \( f \), hence \( fb \) is not in \( T_\Sigma \). Let \( f^{(0)}x \) denote the term \( x \), and for \( i > 0 \), let \( f^{(i)}x \) denote the term \( f f \ldots f x \) where \( f \) occurs \( i \) times.

Let \( \sim \) be the reflexive, symmetric and transitive closure of the relation \( R \) given by:

\[
R_s = \{(a, b), (b, c)\} \cup \{(f^{(i)}a, f^{(i)}c)\}
\]

\[
R_{s_1} = R_{s_2} = R_{s_3} = \emptyset.
\]

Clearly, \( \sim \) is an order-sorted congruence. Let \( A \) denote the quotient \( \Sigma \)-algebra \( T_\Sigma/\sim \). We have \([a] = [b] = [c] = \{a, b, c\}, [f^{(i)}a] = [f^{(i)}c] = \{f^{(i)}a, f^{(i)}c\} \). The carrier of sort \( s_1 \) and \( s_3 \) is \([a], [fa], [ffa], \ldots, [f^{(i)}a], \ldots\), the carrier of sort \( s_2 \) is \([a]\).

How does one define \( f_A^{s_1,s_1}([a]) \)? From one point of view since \( a \in [a] \), we can say \( f_A^{s_1,s_1}([a]) = [fa] = \{fa, fc\} \). But since \([a] = [b] \), we could let \( f_A^{s_1,s_1}([a]) = [fb] \) which is undefined. Definition 2.21 takes care of this since it demands that \( f_A^{s_1,s_1}([a]) = [fa'] \) for some \( a' \) of sort \( s_1 \) such that \( a' \in [a] \). This forces \( a' = a \), and rules \( f_A^{s_1,s_1}([a]) = [fb] \) out. It also rules the choice of \( c \) for \( a' \) since \( c \) is not of sort \( s_1 \). However, \( fc \in [fa] \) implies that \([fa] = [fc] \) hence \( f_A^{s_1,s_1}([a]) = [fc] \) is correct.

### 2.9 The top algebra

As described above, order-sorted congruences are used in defining quotient algebras. The construction of such congruences is somewhat a delicate process because of their order-
2.9 The top algebra

sorted characteristic. If the signature is coherent, one can define a special kind of many-sorted congruence which is simpler to construct and which naturally extends to an order-sorted congruence. This congruence is therefore a useful tool for the construction of order-sorted quotient algebras as shown in section 3.3 where it is used to construct a model. The many-sorted congruence is defined on the top sorts which constitute a many-sorted algebra which we call the top algebra.

Given a coherent order-sorted algebra \( A \), the carriers for the top sorts in \( S \) and the functions defined on those sorts form a many-sorted algebra \( \text{top}(A) \). More formally

**Definition 2.22** Given a coherent order-sorted algebra \( A \) with signature \( (S, \leq, \Sigma, \rho) \), the many-sorted signature \( (S^t, \Sigma^t, \rho^t) \) is defined by considering the top sorts in \( \Sigma \):

\[
S^t = \{ \omega \in S \mid \omega \text{ is a top sort} \}
\]

\[
\Sigma^t = \{ f \in \Sigma \mid \rho(f) \cap (S^t)^* \times S^t \neq \emptyset \}
\]

\[
\rho^t(f) = \rho(f) \cap (S^t)^* \times S^t
\]

Alternatively, one can look at \( \Sigma^t \) as the family \( \Sigma_{(\omega,s)} \) where \( \omega \) and \( s \) are top sorts. From a \((S, \leq, \Sigma, \rho)\)-algebra \( A \) one obtains a \((S^t, \Sigma^t, \rho^t)\)-algebra \( \text{top}(A) \) by restricting \( A \) to the top sorts:

- for \( s \) a top sort, \( \text{top}(A)_s = A_s \)
- for \( f \in \Sigma^t, (\omega, s) \in \rho^t(f) \), \( f^\omega_s = f^\omega \)

The algebra \( \text{top}(A) \) is called the top algebra of \( A \).

**Example 2.4** Consider the signature presented in example 2.1 and the algebra presented in example 2.2. For completeness, we present them once more.

The partially ordered set of sorts is \( S = \{ \text{zero}, \text{rat}^+, \text{rat} \} \) with \( \text{zero} \leq \text{rat}, \text{rat}^+ \leq \text{rat} \).

The signature \( \Sigma \) is:

\[
\Sigma_{\text{rat-rat.rat}} = \{+\}
\]

\[
\Sigma_{\text{rat.rat.rat}^+} = \{/\}
\]

The order-sorted \( \Sigma \)-algebra \( A \) is:

- \( A_{\text{rat}} = Q \) (the set of rational numbers), \( A_{\text{rat}^+} = Q - \{0\} \) (the set of non-zero rationals), and \( A_{\text{zero}} = \{0\} \). The functions have their natural interpretations: + is addition of rational numbers, / is division.

Division is missing from the algebra \( \text{top}(A) \) which is as follows: \( \text{top}(A)_{\text{rat}} = Q \), + is addition. The set \( S^t \) of top sorts is \{\text{rat}\} and the signature of this top algebra is \( \Sigma'_{\text{rat.rat.rat}} = \{+\} \).

**Definition 2.23** We say that a many-sorted congruence \( \sim \) on \( \text{top}(A) \) has the lower sorts property iff \( x_1 \sim_{t_1} y_1, \ldots, x_n \sim_{t_n} y_n \) implies that for any function symbol \( f \in \Sigma \) of rank \((\sigma, \omega)\) where \( \sigma = \sigma_1 \ldots \sigma_n \) with \( \sigma_i \leq t_i \), if \( x_i \) and \( y_i \) are also of sort \( \sigma_i \) then

\[
f^\omega_A(x_1, \ldots, x_n) \sim_{t_i} f^\omega_A(y_1, \ldots, y_n)
\]

for \( \omega_i \) the top sort of the connected component of \( \omega \).
Notice in the above definition that \( f \) is not required to be a member of \( \Sigma' \), the signature of \( \text{top}(A) \). However \( f^\omega_t(x_1, \ldots, x_n) \) is an element of \( A_t \) for every sort \( t \geq \omega \), in particular of \( A_\omega = (\text{top}(A))_\omega \), hence the definition makes sense. Also notice that in the case of initial algebras, by lemma 2.5, if two elements are congruent in some sort, they are congruent in every sort to which both belong. This allows us to strengthen the lower sorts condition on initial order-sorted algebras as follows.

**Lemma 2.6** A congruence \( \sim \) on the top algebra of an initial algebra has the lower sorts property if and only if

\[
x_1 \sim y_1, \ldots, x_n \sim y_n \text{ implies that for any function symbol } f \in \Sigma \text{ such that } f_A(x_1, \ldots, x_n) \text{ and } f_A(y_1, \ldots, y_n) \text{ are defined and are of the same sort, } f_A(x_1, \ldots, x_n) \sim f_A(y_1, \ldots, y_n).
\]

**Example 2.5** On the top algebra of the previous example, define \( x \sim y \) iff \( |x - y| \) is even. Since the only function symbol on \( \text{top}(A) \) is \(+\), clearly \( \sim \) is a \( \text{top}(A) \)-congruence. However, it does not satisfy the lower sorts property. For example \( 1 \sim 3 \) and \( 2 \sim 4 \) but \( 1/2 \not\sim 3/4 \) even though \( 1/2 \) and \( 3/4 \) are in \( \text{top}(A) \).

In this previous example \( \sim \) is not an \( A \)-congruence. As the following lemma shows, this is not a coincidence.

**Lemma 2.7** The restriction of an \( A \)-order-sorted congruence to \( \text{top}(A) \) has the lower sorts property.

**Proof:** Let \( \simeq \) be an order-sorted congruence on \( A \) and let \( \sim \) be its restriction to \( \text{top}(A) \). Suppose \( x_i, y_i \in \text{top}(A)_t \) and \( x_i \sim_t y_i \) for \( 1 \leq i \leq n \). By definition of \( \sim \), it must be the case that \( x_i \simeq_t y_i \) as well. Let \( (\sigma_1 \ldots \sigma_n, \omega) \in \rho(f) \) with \( \sigma_i \leq t_i \). If \( x_i \) and \( y_i \) are also of sort \( \sigma_i \), by the order sorted nature of \( \simeq \), we must have \( x_i \simeq_{\sigma_i} y_i \). Since \( \simeq \) is a congruence, \( f_A(x_1, \ldots, x_n) \simeq_\omega f_A(y_1, \ldots, y_n) \) which by order-sortedness implies \( f_A(x_1, \ldots, x_n) \sim_\omega f_A(y_1, \ldots, y_n) \) for \( \omega_i \) the top element of the connected component of \( \omega \).

The converse is true as well, namely that from a many-sorted congruence on the top algebra one can obtain an order-sorted congruence on the order-sorted algebra.

**Lemma 2.8** Let \( \sim \) be a many-sorted congruence on \( \text{top}(A) \) that satisfies the lower sorts property, then the \( S \)-indexed relation \( \simeq \) defined by

\[
\simeq_s = \{ (x, y) \mid x \sim_t y \text{ for } t \geq s \}
\]

is an \( A \)-order-sorted congruence.

**Proof:** Clearly, \( \simeq \) is an equivalence relation, we show that it is a congruence. Let \( x_i \simeq_{\sigma_i} y_i \) for \( 1 \leq i \leq n \), and let \( f \) be a function symbol with \( (\sigma_1 \ldots \sigma_n, \omega) \) in its rank. We have to show that \( f_A(x_1, \ldots, x_n) \simeq_\omega f_A(y_1, \ldots, y_n) \). Indeed, by the definition of \( \simeq \), \( x_i \sim_{t_i} y_i \) for \( t_i \) the top sort of the connected component of \( \sigma_i \). Let \( \omega_i \) be the top sort in the component of \( \omega \), by the lower sorts property, \( f_A(x_1, \ldots, x_n) \sim_{\omega_i} f_A(y_1, \ldots, y_n) \). Hence by definition \( f_A(x_1, \ldots, x_n) \simeq_\omega f_A(y_1, \ldots, y_n) \) as wanted.

It is easy to see that \( \simeq \) is order-sorted, therefore it is an order-sorted congruence.
These two lemmas actually show the existence of a one to one correspondence between the order-sorted congruences of an order-sorted algebra and the many-sorted congruences on its top algebra satisfying the lower sorts property. In the first lemma, from an order-sorted congruence \( \sim_1 \) one obtains a many-sorted congruence \( \sim_2 \) on the top algebra which satisfies the lower sorts property. Applying the second lemma, an order-sorted congruence \( \sim_3 \) is obtained. Since we are dealing with extensions and restrictions of a relation, it is easy to see that \( \sim_1 \) and \( \sim_3 \) are the same. Similarly if we start from a many-sorted congruence satisfying the lower sorts property, we obtain an order-sorted congruence whose restriction to the top algebra is the congruence with which we started.

This correspondence is useful when one wishes to construct order-sorted congruences since it is simpler to construct a many-sorted congruence for the many sorted top algebra than dealing directly with the order-sorted algebra itself. We use this result in the next section in order to construct an infinite model. Notice that this correspondence is monotonic in the following sense. Let \( \sim_1 \) and \( \sim_2 \) be many-sorted congruences on the top algebra, and let \( \sim_1^* \) and \( \sim_2^* \) be the corresponding order-sorted congruences. Then

\[
\sim_1 \subseteq \sim_2 \quad \text{iff} \quad \sim_1^* \subseteq \sim_2^*
\]

We thus have as a corollary:

**Corollary 2.9** Given a coherent order-sorted signature \( \Sigma \) and a \( \Sigma \)-algebra \( A \), the least many-sorted congruence with the lower sorts property on \( \text{top}(A) \) exists if and only if the least order-sorted congruence on \( A \) exists. Furthermore, if they exist, they correspond to each other under the correspondence described above.

### 2.10 Least Order-sorted congruences

**Definition 2.24** Given an \( S \)-sorted relation \( R \) on an \( < S, \leq, \Sigma, \rho > \)-algebra \( A \), the least order-sorted congruence of \( R \) on \( A \), written \( \sim_R \), is the least (with respect to inclusion) order-sorted congruence containing \( R \).

One way to show the existence of \( \sim_R \) is to realize that it is the intersection of all the order-sorted congruences containing \( R \). Since there is at least one such congruence, namely \( \{(u, v) \mid u, v \in A\} \), and all of them contain \( R \), it is clear that this intersection is not empty. Another way is to give an inductive construction, as for the many-sorted case, and then prove that the congruence is order-sorted.

If the signature is coherent, we can use the results of section 2.9 on top algebras as follows. Let \( R^t \) be the restriction to the top sorts of the completion of \( R \), that is: for a top sort \( \sigma \), \( R^t_\sigma = \{(u, v) \mid (u, v) \in R, \text{ for some } s \leq \sigma\} \). The least congruence containing \( R^t \) with the lower sorts property on \( \text{top}(A) \) induces, by virtue of corollary 2.9, the least order-sorted congruence \( \simeq_R \) of \( R \) on \( A \).

Thus we proceed to show how we can construct \( \simeq_Q \), the least congruence of a relation \( Q \) with the lower sorts property on \( \text{top}(A) \). In our case \( Q = R^t \). For a relation \( Q \), let \( Q^{-1} = \{(v, u) \mid (u, v) \in Q\} \). The construction is in stages as follows:
Construction 2.10

\[ \simeq_{0,s} = \{(u, u) \mid u \in \text{top}(A)_s\} \cup Q_s \cup Q_s^{-1} \]

for \( i > 0 \),

\[ \simeq_{i,s} = \simeq_{i-1,s} \]
\[ \cup \{(u, w) \mid \exists v (u \simeq_{i-1,s} v \simeq_{i-1,s} w)\} \]
\[ \cup \{(f_A(u_1, \ldots, u_n), f_A(v_1, \ldots, v_n)) \mid f \in \Sigma, (\sigma_1 \ldots \sigma_n, \omega) \in \rho(f), \omega \leq s, \]
and for \( 1 \leq j \leq n, u_j \simeq_{i-1,\sigma_j} v_j\}\}

Notice that we use the functions \( f_A \) (on \( A \)) and not \( f_{\text{top}}(A) \) (on \( \text{top}(A) \)), this is what ensures the lower sorts property. The third member of the union can be more informally expressed as

\[ \{(f_A(u_1, \ldots, u_n), f_A(v_1, \ldots, v_n)) \mid f_A(u_1, \ldots, u_n) \text{ and } f_A(v_1, \ldots, v_n) \]
are well defined and for \( 1 \leq j \leq n, u_j \simeq_{i-1,\sigma_j} v_j \text{ for some sort } \sigma_j\}\}

Finally, let \( \simeq = \bigcup_{i \geq 0} \simeq_i \).

One can see that \( \simeq \) is a many-sorted congruence on \( \text{top}(A) \). It is clearly reflexive, symmetry follows from the fact that \( Q \) and \( Q^{-1} \) are in \( \simeq_0 \), and transitivity is ensured by the second member of the union in the construction of \( \simeq_i \). Thus \( \simeq \) is an equivalence relation. Furthermore \( \simeq \) is preserved under function application by virtue of the third member of this union. Actually in that case we are adding more than is needed for a simple \( \text{top}(A) \) congruence, namely we are causing it to satisfy the lower sorts property.

That \( \simeq \) is minimal can be shown by inductively proving \( \simeq_i \subseteq R \) for any other many-sorted congruence \( R \) with the lower sorts property. Therefore \( \simeq \) is the least congruence on \( \text{top}(A) \) with the lower sorts property containing \( Q \).

What are the ways in which two elements become congruent? It is clearly by propagation of \( Q \) and \( Q^{-1} \) via transitivity and function application. Thus whenever \( u \simeq v \) there should be a way of specifying exactly which pairs in \( Q \) have been used. We now formalize this intuition.

Definition 2.25 Based on the least congruence \( \simeq \), the relation \( \sim \) is defined as follows:

\[ u \sim_0 v \text{ if and only if } u = v \]
\[ \text{and for } i \geq 0, \]
\[ u \sim_{i+1} v \text{ if and only if either } u = v \text{ or } \]
\[ u = f_A(u_1, \ldots, u_n), v = f_A(v_1, \ldots, v_n), \text{ and for } 1 \leq j \leq n, u_j \simeq_{i,\sigma_j} v_j. \]

Notice that \( u \sim_i v \) implies \( u \simeq_v v \), hence \( \sim_i \) is a weaker notion. Also, \( u \sim_i v \) implies \( u \sim_{i+1} v \).

Lemma 2.11 If \( u \sim_i v \sim_i w \) then \( u \sim_{i+1} w \).
Proof: If either \( u = v \) or \( v = w \), the claim clearly holds. Suppose \( u = v \), then \( v \sim v \).
2.10 Least Order-sorted congruences

implies \( v \sim_{i+1} w \). Similarly for \( v = w \).

Otherwise, \( u = f_A(u_1, \ldots, u_n), v = f_A(v_1, \ldots, v_n), w = f_A(w_1, \ldots, w_n) \) and for \( 1 \leq j \leq n \), \( u_j \sim_{i-1} v_j \sim_{i-1} w_j \). By definition of \( \sim_i \), we have \( u_j \sim_i w_j \), which forces \( u = f_A(u_1, \ldots, u_n) \sim_{i+1} f_A(w_1, \ldots, w_n) = w \) as wanted.

We are now ready to prove the lemma that relates two congruent terms to the pairs in \( Q \) which make them congruent.

**Lemma 2.12** If \( u \sim_i v \), then either

1. \( u = v \) or,
2. \( u \sim_i v \) (and \( u \neq v \)) or,
3. There exist \((l_1, r_1), \ldots, (l_n, r_n)\) in \( Q \cup Q^{-1} \) such that
   \( u \sim_i l_1, \ldots, r_j \sim_i l_{j+1}, \ldots, r_n \sim_i v \).

**Proof:** The claim is clearly true for \( \sim_0 \) for either \( u = v \) or \( (u, v) \in Q \cup Q^{-1} \) in which case \( l_1 = u, r_1 = v \) and \( u \sim_0 l_1, r_1 \sim_0 v \), so that \( (u, v) \) is a chain of length 1 that satisfies 3.

Assume that the claim holds for \( 0 \leq k < i \). Suppose that \( u \sim_i v \) and that \( i \) is minimal. There are two ways in which \( u \) and \( v \) can become congruent:

1. Function application: for \( 1 \leq j \leq n \) \( u_j \sim_{i-1} v_j \) and \( u = f_A(u_1, \ldots, u_n), v = f_A(v_1, \ldots, v_n) \). Therefore \( u \sim_i v \); or
2. Transitivity: there exists \( y \) such that \( u \sim_{i-1} y \sim_i v \). By inductive hypothesis, we can say that one of the following cases occurs for \( u \) and \( y \):
   a1) \( u = y \) or
   a2) \( u \sim_{i-1} y \) or
   a3) \( u \sim_{i-1} l_1, r_1 \sim_{i-1} l_2, \ldots, r_n \sim_{i-1} y \) for some \((l_1^u, r_1^u), \ldots, (l_n^u, r_n^u)\) in \( Q \cup Q^{-1} \).

Similarly \( y \sim_{i-1} v \) implies that
   b1) \( y = v \) or
   b2) \( y \sim_{i-1} v \) or
   b3) \( y \sim_{i-1} l_1^v, r_1 \sim_{i-1} l_2^v, \ldots, r_n \sim_{i-1} v \) for some \((l_1^v, r_1^v), \ldots, (l_n^v, r_n^v)\) in \( Q \cup Q^{-1} \).

Case a1 would violate the minimality condition as would case b1. If case a2 and b2 occur then, by lemma 2.11 we have that \( u \sim_{i-1} y \sim_{i-1} v \), that is \( u \sim_i v \).

Suppose now that a2 and b3 occur. Then \( u \sim_{i-1} y \sim_{i-1} l_1^v \). By lemma 2.11 we have \( u \sim_i l_1^v, r_1 \sim_i l_2^v, \ldots, r_n \sim_i v \) as wanted.

The case a3 and b3 is handled similarly. Notice that \( r_{n_u}^u \sim_{i-1} y \sim_{i-1} l_1^v \), hence \( r_{n_u}^u \sim_i l_1^v \), and \((l_1^v, r_1^v), \ldots, (l_n^v, r_n^v)\) are such that \( u \sim_i l_1^v, r_1 \sim_i l_2^v, \ldots, r_n \sim_i v \) as wanted.

\[ \square \]
3 Congruences Associated With Order-sorted Horn Clauses

Let \( (S, \leq, \Sigma, \rho) \) be a coherent signature, and let \( T = T_{(S, \leq, \Sigma, \rho)} \) be the order-sorted term algebra on that signature. Let \( H_1 \) be a set of ground order-sorted Horn clauses over \( T \), possibly with equational atoms, and let \( H \) be the set of equational Horn clauses obtained as described in section 2.6.

3.1 The Graph \( GT(H) \)

In order to obtain efficiency and to avoid empty carriers, we restrict our attention to the sorts which actually appear in the set of clauses, as opposed to the set of all sorts in the logic language to which the clauses belong.

**Definition 3.1** Given a set \( H \) of ground Horn clauses over the term algebra of a coherent order-sorted signature \( (S, \leq, \Sigma, \rho) \), let

- \( \text{TERM}(H) \) be the set of all subterms of terms occurring in the atomic formulae in \( H \); and
- \( S(H) \) be the set of sorts of all terms in \( \text{TERM}(H) \), partially ordered by the restriction of \( \leq \) to \( S(H) \).

The graph \( GT(H) \) represents subterm dependencies, and it is used to propagate congruential information. This graph was first defined by Kozen (under a different name) to study the properties of finitely presented algebras, \([8,9,10,11]\).

For every sort \( s \) in \( S(H) \), let \( \text{TERM}(H)_s \) be the set of all terms of sort \( s \) in \( \text{TERM}(H) \). Note that by the definition, each set \( \text{TERM}(H)_s \) is nonempty. Let \( \Sigma(H) \) be the \( S(H) \)-ranked alphabet consisting of all constant and function symbols occurring in \( \text{TERM}(H) \) and let \( \rho(H) \) be the restriction of \( \rho \) to \( \Sigma(H) \).

Consider the signature \( (S(H), \leq, \Sigma(H), \rho(H)) \). Notice that \( \Sigma(H) \subseteq \Sigma \) and if \( \Sigma \) is coherent, so is \( \Sigma(H) \).

The graph \( GT(H) \) has the set \( \text{TERM}(H) \) as its set of nodes, its edges and the function \( \Lambda \) labeling its nodes are defined as follows:

- For every node \( t \) in \( \text{TERM}(H) \), if \( t \) is a constant, then \( \Lambda(t) = t \), else \( t \) is of the form \( fy_1 \ldots y_k \) and \( \Lambda(t) = f \);

- For every node \( t \) in \( \text{TERM}(H) \), if \( t \) is of the form \( fy_1 \ldots y_k \), then \( t \) has exactly \( k \) successors \( y_1 \ldots y_k \), else \( t \) is a constant and it is a terminal node of \( GT(H) \).

Given a node \( u \in \text{TERM}(H) \), if \( u \) has \( n \) successors, \( n > 0 \), then the \( i \)-th successor of \( u \) is denoted by \( u[i] \).
3.2 Order-sorted Graph Congruence Closure

Example 3.1 Consider the signature in which \( S = \{i, s\} \), \( \Sigma = \{f, g, a, b, c\} \), with \( \rho(f) = (is, s) \), \( \rho(g) = (si, s) \), \( \rho(a) = i \), \( \rho(b) = \rho(c) = s \). Let \( H \) be the set of Horn clauses

\[ \{f(a, b) \doteq c, -g(f(a, b), a) \doteq g(c, a)\} \]

Then, \( TERM(H)_i = \{a\} \), \( TERM(H)_s = \{b, c, f(a, b), g(c, a), g(f(a, b), a)\} \), and \( E = \{(f(a, b), c)\} \). The graph \( GT(H) \) is shown in figure 2.

3.2 Order-sorted Graph Congruence Closure

The crucial concept in showing the decidability of unsatisfiability for ground equational Horn clauses is a certain kind of equivalence relation on the graph \( GT(H) \) called an order-sorted graph congruence.

Definition 3.2 Given the graph \( GT(H) \) associated with the set \( H \) of ground Horn clauses, an \( S(H) \)-indexed family \( R \) of relations \( R_s \) over \( TERM(H)_s \) is an order-sorted graph congruence on \( GT(H) \) iff:

1. Each \( R_s \) is an equivalence relation;
2. For every pair \( (u, v) \in TERM(H)^2_s \), if \( \Lambda(u) = \Lambda(v), (w_1 \ldots w_n, s) \in \rho(\Lambda(u)) \), and for every \( i, \ 1 \leq i \leq n \), \( u[i] R_w v[i] \), then \( u R'_s v \).
3. For every pair \( (u, v) \in TERM(H)^2_s \), if \( s_1 \in co-arity(\Lambda(u)) \cap co-arity(\Lambda(v)) \) and \( s_1 \leq s_2 \) then \( u R_{s_1} v \iff u R_{s_2} v \).
4. For every pair \( (u, v) \) of nodes in \( TERM(H)^2_s \):
   - If \( u \doteq v \in H \), then \( u R_s v \); and
   - if \( u \doteq v \) is the head of a clause \( u \doteq v : -u_1 \doteq v_1, \ldots, u_n \doteq v_n \) in \( H \), and for every \( i, \ 1 \leq i \leq n \), \( u_i R_{s_i} v_i \), then \( u R_s v \).
Given an $S(H)$ indexed family of relations $E_s$ on $TERM(H)$, the order-sorted graph congruence closure of $E$, denoted $\cong_E$, is the least order-sorted graph congruence on $GT(H)$ containing $E$. In section 4 we prove the existence of $\cong_E$ and provide a construction for it.

We now concentrate on the relationship between the graph congruence closure and the least order-sorted congruence. Let $\Sigma(H)$ be the set of all function and constant symbols appearing in $H$. Consider the (free) order-sorted $\Sigma(H)$-term algebra $\mathcal{T}_H$. Every function symbol is interpreted as a term constructor. Notice that $TERM(H) \subseteq \mathcal{T}_H$ and $\cong_E$ is a graph congruence which is not necessarily a $\mathcal{T}_H$-congruence, mainly because $TERM(H)$ is finite and not closed under function application. If we want to make $\cong_E$ into an order-sorted-$\Sigma(H)$ congruence we can consider $\cong$ (which we denote by $\sim$), the least order-sorted congruence containing $\cong_E$. What is the relationship between $\cong_E$ and $\sim$? As we now show, $\sim$ is a conservative extension of $\cong_E$.

**Lemma 3.1** [Conservativeness] For any $u, v \in TERM(H)$, $u \cong_E v$ if and only if $u \sim v$.

**Proof:** Recall that $\sim$ was constructed in stages (construction 2.10 on page 14). We show by induction on $i$ that for any $u, v \in TERM(H)$, $u \sim_i v$ implies $u \cong_E v$.

This is clearly the case for $i = 0$.

Let $i > 0$ be the least number satisfying $u \sim_i v$. By lemma 2.12 we have one of the following cases:

1. $u = v$ which cannot occur because of the minimality of $i$, or
2. $u \sim_i v$. Then $u = f(u_1, \ldots, u_n)$, $v = f(v_1, \ldots, v_n)$ and for $1 \leq j \leq n$, $u_j \sim_{i-1} v_j$.
   Since $TERM(H)$ is closed under subterms, $u_j, v_j \in TERM(H)$ and the inductive hypothesis can be applied to conclude $u_j \cong_E v_j$. Since $\cong_E$ is closed under function application we have: $u = f(u_1, \ldots, u_n) \cong_E v = f(v_1, \ldots, v_n)$;
3. There exist $(l_1, r_1), \ldots, (l_n, r_n) \in \cong_E$ such that $u \sim_i l_1 \cong_E r_1 \sim_i l_2 \cong_E r_2$, \ldots, $r_j \sim_i l_{j+1} \cong_E r_{j+1}$, \ldots, $l_n \cong_E r_n \sim_i v$. As in the previous case, since $u$ and $l_l$ are in $TERMS(H)$ and $u \sim_i l_1$, we can prove $u \cong_E l_1$. The same argument applies to show that $r_j \cong_E l_{j+1}$ for $1 \leq j < n$ and $r_n \cong_E v$. Hence we have that $u \cong_E l_1 \cong_E \ldots \cong_E r_n \cong_E v$. Since $\cong_E$ is an equivalence relation it is transitively closed, and therefore $u \cong_E v$.

A more formal treatment of this case would use and inductive argument on pairs $< i, n >$ ordered lexicographically.

\[ \square \]

### 3.3 A Method For Testing Unsatisfiability

For every top sort $s \in S(H)$, let $E_s$ be the set of all pairs $(l, r)$ such that $s$ is the top sort of the connected component to which $l$ and $r$ belong and $l \equiv r$ is an equation in $H$. Notice that the above definition makes sense because $l \equiv r$ can occur in $H$ provided the sorts of $l$ and $r$ have a common element. Since the signature is coherent, all the sorts of a term
are in one connected component, hence all the sorts of \( l \) and \( r \) are in the same connected component. Let \( E \) be the \( S(H) \)-indexed family \((E_s)_{s \in S(H)}\). The key to the method is that \( \leftrightarrow_E \), the order-sorted congruence closure of \( E \), exists and that there is an algorithm for computing it.

The proof is somewhat more complex than the one given by Gallier in [2] where a finite model is presented. There, a finite algebra is constructed from the elements in \( TERM(H) \). The interpretation of a function symbol \( f \) of rank \( \rho(f) = (\sigma_1 \ldots \sigma_n, s) \) on \( x_1, \ldots, x_n \) (where \( x_i \) is of sort \( \sigma_i \)), is defined to be either

1. \( f x_1 \ldots x_n \) if \( f x_1 \ldots x_n \in TERM(H) \), or

2. \( f z_1 \ldots z_n \) if \( \exists z_1, \ldots, z_n \) such that for \( 1 \leq j \leq n, x_j \leftrightarrow_E z_j \) and \( f z_1 \ldots z_n \in TERM(H) \), or

3. \( c \) for some element \( c \in TERM(H)_s \), otherwise.

In the order-sorted case, case 3 causes trouble as can be seen in the following example.

**Example 3.2** Let the sort structure be: \( s_1, s_2 \leq s \) and \( t_1 \leq t \). Suppose \( \Sigma \) consists of the constants \( a \) of sort \( s_1 \), \( b \) of sort \( s_2 \) and \( c \) of sort \( t_1 \), in addition there is a function symbol \( f \) of rank \( \rho(f) = ((s, t), (s_2, t_1)) \). Figure 3 depicts the situation.

Let the set \( H \) of Horn clauses be \( \{a \equiv b, fa \equiv fa, c \equiv c\} \). The formulae \( fa \equiv fa \) and \( c \equiv c \) are present just to force \( fa \) and \( c \) to be in \( TERM(H) \).

Since \( fb \notin TERM(H) \) and \( c \) is the only element of sort \( t_1 \), we would define \( f_M^{s_2 t_1}(b) = c \). Hence by order-sortedness \( f_M^{s_2 t_1}(b) = c \) as well.

On the other hand, since \( fa \in TERM(H) \), \( f_M^{s_1 t}(a) = fa \). Since \( a \equiv b \), \( f_M^{s_1 t}(a) \) and \( f_M^{s_1 t}(b) \) would have to be congruent, i.e. \( fa \) congruent to \( c \). However, clearly \( fa \) and \( c \) are not congruent modulo \( \leftrightarrow_E \).

Our construction is therefore different in that we deal with an infinite model containing \( TERM(H) \) and a congruence which is conservative over \( \leftrightarrow_E \). From the infinite model we present, one can then extract a finite one.
Theorem 3.2 [Soundness and completeness] Let $H$ be a set of order-sorted ground Horn clauses (with equality), and let $E$ be as defined above. If $\leftrightarrow_E$ is the order-sorted congruence closure on $GT(H)$ of $E$, then

\[
H \text{ is unsatisfiable if and only if for some clause } - u_1 = v_1, \ldots, u_n = v_n \text{ in } H,
\]

for every $i$, $1 \leq i \leq n$, we have $u_i \leftrightarrow_E v_i$.

**Proof:** The proof is obtained by combining and generalizing the techniques used by Gallier in [Z]. Let $D$ be the subset of $H$ consisting of the set of definite clauses in $H$.

First, we show that the $S(H)$-indexed family $R$ of relations $R_s$ on $TERM(H)$ defined such that $t R_s u \iff D \models t \equiv u$, is a congruence on $GT(H)$ containing $E$. The details are straightforward and are left to the reader.

Since $\leftrightarrow_E$ is the least congruence on $GT(H)$ containing $E$, for any terms $l, r \in TERM(H)$, we have $l \leftrightarrow_E r$, then $D \models l \equiv r$.

Then, if for some negative clause $- u_1 = v_1, \ldots, u_n = v_n$ in $H$, we have $u_i \leftrightarrow_E v_i$ for every $i$, $1 \leq i \leq n$, then $D \models u_1 = v_1 \land \ldots \land u_n = v_n$ holds, which implies that the set $D \cup \{- u_1 = v_1, \ldots, u_n = v_n\}$ is unsatisfiable. Consequently, $H$ is unsatisfiable. Notice that as a matter of fact only the top sort components of $\leftrightarrow_E$ are needed here since every $l$ and $r$ belong to some top sort $t$ and $l \leftrightarrow_{E,t} r$ iff $l \leftrightarrow_{E,t} r$.

Conversely, assume that there is no negative clause $- u_1 = v_1, \ldots, u_n = v_n$ in $H$ such that, $u_i \leftrightarrow_E v_i$ for every $i$, $1 \leq i \leq n$. We shall construct a model $M$ of $H$.

Let $\Sigma_1$ be the set of all function and constant symbols excluding the ones which represent predicate letters (notice that $\Sigma_1 \subseteq \Sigma$). Consider the initial order-sorted $\Sigma_1$-term algebra $T_{\Sigma_1}$. Every function symbol is interpreted as a term constructor. We will construct the model $M$ by first defining a congruence on $T_{\Sigma_1}$ and then giving an interpretation for the predicate symbols on the quotient algebra.

Let $\Sigma_1$ be the set of all function and constant symbols excluding the ones which represent predicate letters (notice that $\Sigma_1 \subseteq \Sigma$). Consider the initial order-sorted $\Sigma_1$-term algebra $T_{\Sigma_1}$. Every function symbol is interpreted as a term constructor. We will construct the model $M$ by first defining a congruence on $T_{\Sigma_1}$ and then giving an interpretation for the predicate symbols on the quotient algebra.

Let $\sim$ be the least order-sorted congruence containing $\leftrightarrow_E$ on $T_{\Sigma_1}$. By the conservativity lemma 3.1, we have that for $u, v \in TERM(H)$ of sort $s \neq bool$,

\[
\leftrightarrow_{E,s} \iff u \cong_s v.
\] (1)

This property is important because it shows that incongruences on the graph are preserved by the algebra congruence. This is needed to show that our algebra is indeed a model.

Consider the quotient $T_{\Sigma_1} / \sim$ which is itself an order-sorted algebra. Notice that (1) implies that for nodes $u, v$ of sort $s \neq bool$ in $TERM(H)$,

\[
[u]_\sim = [v]_\sim \iff u \leftrightarrow_E v.
\] (2)
We now provide an interpretation for the predicate symbols, thereby making the $\Sigma_1$ algebra $T_{\Sigma_1}/\simeq$ into a $\Sigma$-algebra $\mathcal{M}$. The symbol $\top$ is interpreted as $\text{true}$. For every predicate symbol $P$, if $(\sigma_1 \ldots \sigma_n) \in \rho(f)$, and $[u_1], \ldots, [u_n]$ are of sort $\sigma_1 \ldots \sigma_n$, then

$$P_M([u_1], \ldots, [u_n]) \text{ def } \begin{cases} \text{true} & \text{iff for every } i, \ 1 \leq i \leq n \text{ there exists a } z_i \in [u_i] \\ \text{s.t. } Pz_1 \ldots z_n \xrightarrow{\ E, \text{bool} \ \top} & \\ \text{false} & \text{otherwise} \end{cases}$$

It is obvious from the definition that if $P$ is of ranks $(\tau, \text{bool})$ and $(\sigma, \text{bool})$ with $\sigma \leq \tau$, then

$$P_M^x([u_1], \ldots, [u_n]) = \text{true} \text{ iff } P_M^{x, \text{bool}}([u_1], \ldots, [u_n]) = \text{true}. \text{ Hence this is an order-sorted } \Sigma \text{ algebra, which we denote by } \mathcal{M}. \text{ The graph congruence is preserved in } \mathcal{M}, \text{ i.e. given terms } u_1, \ldots, u_n \text{ in } \text{TERM}(H),$$

$$P_M([u_1], \ldots, [u_n]) = \text{true} \text{ iff } Pu_1 \ldots u_n \xrightarrow{E, \text{bool} \ \top}.$$

Clearly, $Pu_1 \ldots u_n \xrightarrow{E, \text{bool} \ \top}$ implies $P_M([u_1], \ldots, [u_n]) = \text{true}$ since $u_i \in [u_i]$. For the converse, assume $P_M([u_1], \ldots, [u_n]) = \text{true}$. Then there exist $z_1, \ldots, z_n$ such that $z_i \simeq u_i$ and $Pz_1 \ldots z_n \xrightarrow{E, \text{bool} \ \top}$. Since $u_i$ and $z_i$ are in $\text{TERM}(H)$, by property 3 above we conclude that $u_i \xrightarrow{E} z_i$. Since $\xrightarrow{E}$ is a congruence, this implies $Pu_1 \ldots u_n \xrightarrow{E} Pz_1 \ldots z_n \xrightarrow{E} \top$.

If every non-boolean constant $c$ is interpreted as $[c]$, it can easily be seen that every non-boolean term $t$ is mapped onto $[t]$. Every boolean term is mapped into $\text{true}$ or $\text{false}$. By 3 and 4 above we have:

$$M \models u \equiv v \iff u \xrightarrow{E} v.$$  (5)

We now prove that $\mathcal{M}$ with this interpretation is a model of $H$.

For every clause $u \equiv v \in H$, we have $(u, v) \in E_s$ for the top sort $s$ of the component to which the sorts of $u$ and $v$ belong. Since $\xrightarrow{E}$ is a congruence containing $E$, we have $u \xrightarrow{E} v$. But then, by 5, we have $\mathcal{M} \models u \equiv v$.

For every clause $u \equiv v : \neg u_1 \equiv v_1, \ldots, u_n \equiv v_n$ in $H$, if $\mathcal{M} \models u_i \equiv v_i$ for every $i$, $1 \leq i \leq n$, then, by 5, we have $u_i \xrightarrow{E} v_i$ for $1 \leq i \leq n$. Since $\xrightarrow{E}$ is a congruence on $\text{GT}(H)$, we have $u \xrightarrow{E} v$. By 5, this is equivalent to $\mathcal{M} \models u \equiv v$. Hence,

$$\mathcal{M} \models \neg u_1 \equiv v_1, \ldots, u_n \equiv v_n.$$  (6)

Finally, given any negative clause $: \neg u_1 \equiv v_1, \ldots, u_n \equiv v_n$ in $H$, recall that it is assumed that we cannot have $u_i \xrightarrow{E} v_i$ for every $i$, $1 \leq i \leq n$. Then, for some $j$, $1 \leq j \leq n$, $u_j$ and $v_j$ are not $\xrightarrow{E}$ congruent, and by 5, this implies that $\mathcal{M} \not\models u_j \equiv v_j$, that is, $\mathcal{M} \not\models \neg u_j \equiv v_j$. But this implies that

$$\mathcal{M} \models : \neg u_1 \equiv v_1, \ldots, u_n \equiv v_n.$$  (7)

Hence, $\mathcal{M}$ is a model of every clause in $H$. This concludes the proof.  \hspace{1cm} \Box

It is interesting to note that the soundness part of theorem 3.2 follows from the fact that $\xrightarrow{E}$ is the least congruence on $\text{GT}(H)$ containing $E$, and that the completeness part follows from the fact that $\xrightarrow{E}$ is a graph congruence. It only remains to prove that $\xrightarrow{E}$ exists and to give an algorithm for computing it.
Example 3.3 Consider the signature of example 2.3 on page 10 where the sort structure is $S = \{s_1, s_2, s_3, s\}$ with $s_1, s_2, s_3 \leq s$, $\Sigma = \{a, b, c, f\}$, $a, b, c$ are constants of rank $s_1, s_2$ and $s_3$ respectively, and $f$ has rank $\{(s_1, s_1), (s_3, s_3)\}$. Notice that the signature is coherent and has one connected component: $\{s_1, s_2, s_3, s\}$. Given $H = \{a \doteq b, b \doteq c\}$. It turns out that $fa \doteq fc$ is not a consequence of $H$, the reason being that for any $\Sigma$-algebra $A$, the $f$ appearing in $fa$ and that appearing on $fc$ do not denote the same function. The first denotes a function from $A, \rightarrow A, \rightarrow A$, while the latter a function from $A, \rightarrow A, \rightarrow A$. To disprove $fa \doteq fc$, construct the set $H' = \{a \doteq b, b \doteq c, -fa \doteq fc\}$ and show it is satisfiable. The model $M$ is given by theorem 3.2. The congruence classes are as follows:

- $[a] = [b] = [c] = \{a, b, c\}$
- Let $f^{(0)}x$ denote the term $x$, and for $i > 0$, let $f^{(i)}x$ denote the term $ff\ldots fx$ where $f$ occurs $i$ times. Then $[f^{(i)}a] = \{f^{(i)}a\}$ and $[f^{(i)}c] = \{f^{(i)}c\}$.

The carriers are:

- $M_s = \bigcup_{i \geq 0} \{[f^{(i)}a]\} \cup \{[f^{(i)}c]\}$
- $M_{s_1} = \bigcup_{i \geq 0} \{[f^{(i)}a]\}$
- $M_{s_2} = \{[a]\}$
- $M_{s_3} = \bigcup_{i \geq 0} \{[f^{(i)}c]\}$

The function symbol $f$ is interpreted as $f_M([x]) = [fx]$ for $x$ of sort $s_1$ or $s_3$ (notice that these are two different functions). $M$ satisfies $H'$ because $fa$ and $fc$ are in different congruence classes.

3.4 Finite models

In the last example, the model for $H'$ is infinite. In a sense it is too big. For example, the congruence class of $f^{100}a$ will never be used in proving $\neg fa \doteq fc$.

We pointed out earlier certain difficulties in constructing a finite model. Using $M$, the infinite model of theorem 3.2, we are now in a position to construct a finite model.

For every top sort $t$ a representative element $c_t$ is chosen from $M_t$ (note that $c_t$ is an equivalence class). These elements are to be used for the value of $f[t]$ whenever $ft \notin TERM(H)$. Given a sort $s$, let $top(s)$ denote the top sort of its connected component.

The finite model $M'$ is defined as follows. The carrier for the sort $bool$ and the boolean functions are as in $M$. For a sort $s \neq bool$, its carrier is defined by

$$M'_s = \{[u] \in M_s \mid [u] \cap TERM(H)_s \neq \emptyset\} \cup \{c_{top(s)}\}.$$  

Note that $M'_s$ has at most as many elements as $TERM(H)_s$, hence each carrier is finite.

The interpretation for the function symbols uses the representative elements as follows. Let $\sigma = \sigma_1 \ldots \sigma_n$, given a function symbol $f$ of rank $(\sigma, s) (s \neq bool)$ and terms $t_1, \ldots, t_n$ of sort $\sigma_1, \ldots, \sigma_n$,

$$f^\sigma_M([t_1], \ldots, [t_n]) \overset{df}{=} \begin{cases} [ft_1 \ldots t_n] & \text{if } [ft_1 \ldots t_n] \cap TERM(H)_s \neq \emptyset, \\ c_{top(s)} & \text{otherwise} \end{cases}$$

We now have the following lemma.
Lemma 3.3 If $H$ is satisfiable then $M'$ is a finite model for it.

Notice that $M'$ is not unique since it depends upon the choice of the elements $c_i$.

Example 3.4 Consider the previous example. A finite model $M'$ for $H' = \{a = b, b = c, \neg fa = a\}$ is given as follows. First, choose $[fa]$ to be the representative for $M'_s$, i.e. $c_s = [fa]$. Then add $[fa]$ to all the carriers of sorts below $s$ to obtain:

$$
M_s = \{[a], [fa], [fc]\}
M_{s_1} = \{[a], [fa]\}
M_{s_2} = \{[a], [fa]\}
M_{s_3} = \{[a], [fa], [fc]\}
$$

The functions $f_{M'}$ are therefore given by:

$$
\begin{align*}
f_{M',s_1}^{a}[a] &= [fa] \\
f_{M',s_1}^{a}[fa] &= [fa] \\
f_{M',s_2}^{a}[a] &= [fa] \\
f_{M',s_2}^{a}[fa] &= c_s = [fa] \\
f_{M',s_3}^{a}[fc] &= [fc]
\end{align*}
$$

4 Existence of the Order-sorted Graph Congruence Closure

We now prove that the order-sorted graph congruence closure of a relation $R$ on the graph $GT(H)$ exists. This can be done by interleaving steps in which a purely equational congruence closure is computed, and steps in which a purely implicational kind of closure is computed. The advantage of this method (even though it is not the most direct) is that it justifies the correctness of the algorithm for computing the graph congruence closure of $R$ on $GT(H)$.

First, we define the concept of an equational order-sorted congruence closure.

4.1 Equational Order-sorted Graph Congruence Closure

The notion of many-sorted equational congruence closure was first introduced (under a different name) by Kozen, [8,9]. In fact, Dexter Kozen appears to have given an $O(n^2)$-time algorithm solving the word problem for finitely presented algebras before everyone else [8]. Independently, the concept of congruence closure was defined in Nelson and Oppen, [12]. We have added the qualifier equational in order to distinguish it from the more general notion defined in Section 3.2, that applies to Horn clauses.

For our purpose, we only need to consider the concept of equational order-sorted closure on the graph $GT(H)$ induced by some (fixed) set $H$ of ground Horn clauses. In the rest of this section, it is assumed that a fixed set $H$ of ground Horn clauses is given.
**Definition 4.1** An $S(H)$-indexed family $R$ of relations $R_s$ over $TERM(H)$, is an equational congruence on $GT(H)$ iff:

1. Each $R_s$ is an equivalence relation;
2. For every pair $(u,v) \in TERM(H)^2$, if $\Lambda(u) = \Lambda(v)$, $(w_1 \ldots w_n, s) \in \rho(\Lambda(u))$, and for every $i$, $1 \leq i \leq n$, $u[i]R_w v[i]$, then $uR_s v$.
3. For every pair $(u,v) \in TERM(H)^2$, if $s_1 \in co-arity(\Lambda(u)) \cap co-arity(\Lambda(v))$ and $s_1 \leq s_2$ then $uR_{s_1} v$ iff $uR_{s_2} v$.

A non order-sorted version of the following lemma was first shown by Kozen, [8,9]. For the sake of completeness our proof is based upon the one presented in Gallier, [2,3].

**Lemma 4.1** Given any $S(H)$-indexed family $R$ of relations on $TERM(H)$, there is a smallest equational order-sorted congruence $\hat{R}$ on the graph $GT(H)$ containing $R$.

**Proof:** Since $R$ might not be an order-sorted relation, we complete it to $R'$ (as in definition 2.18):

$$R'_s = \{(u, v) \in A_s \times A_s \mid (u, v) \in R_t \text{ for some } t \text{ such that } t \geq s \text{ or } s \geq t\}.$$

We define the sequence $R^i$ of $S(H)$-indexed families of relations inductively as follows: For every sort $s \in S(H)$, for every $i \geq 0$,

$$R^0_s = R'_s \cup \{(u, u) \mid u \in TERM(H)_s\},$$

$$R^{i+1}_s = R^i_s \cup \{(v, u) \in TERM(H)^2_s \mid (u, v) \in R^i_s\} \cup \{(u, w) \in TERM(H)^2_s \mid \exists v \in TERM(H), (u, v) \in R^i_s \text{ and } (v, w) \in R^i_s\} \cup \{(u, v) \in TERM(H)^2_s \mid \Lambda(u) = \Lambda(v), (w_1 \ldots w_n, s) \in \rho(\Lambda(u)),$$

$$\text{and } u[j]R^i_{w'_j} v[j], 1 \leq j \leq n \text{, for some } s' \text{ connected to } s$$

$$\text{such that } (w'_1 \ldots w'_n, s') \in \rho(\Lambda(u))\}.$$

Let $(\hat{R})_s = \bigcup_{i \geq 0} R^i_s$. Clearly, $\hat{R}$ is a congruence. It is also order-sorted due the use of $R'$ in $R^0$ and the fact that the third union in the definition of $R^{i+1}_s$ deals with a sort connected to $s$. Since $s_1 \leq s_2$ implies that $s_1$ and $s_2$ are connected, any $s'$ is connected to $s_1$ if and only if it is connected to $s_2$. It is easily shown by induction that every equational order-sorted congruence on $GT(H)$ containing $R$ contains every $R^i$, and that $\hat{R}$ is an equational order-sorted congruence on $GT(H)$. Hence, $\hat{R}$ is the least equational order-sorted congruence on $GT(H)$ containing $R$. 

Since the graph $GT(H)$ is finite, there must exist some integer $i$ such that $R^i = R^{i+1}$. Hence, the equational order-sorted congruence closure $\hat{R}$ of $R$ is computable.
4.2 Implicational Closure

Let $H$ be a set of equational ground Horn clauses.

**Definition 4.2** An $S(H)$-indexed family $R$ of relations $R_s$ over $TERM(H)$ is an implicational relation on $GT(H)$ iff: For every pair $(u, v)$ of nodes in $TERM(H)^2$:

1. If $u = v \in H$, then $uR_sv$ for each sort $s$ such that $u$ and $v$ are of sort $s$.
2. If $u = v$ is the head of a clause $u \doteqdot v : - u_1 \doteqdot v_1, \ldots, u_n \doteqdot v_n$ in $H$, and for every $j$, $1 \leq j \leq n$, there exists a sort $w_j$ such that $u_j$ and $v_j$ are of sort $w_j$ and $u_jR_{w_j}v_j$, then $uR_sv$.

The following result is well known, (for instance, see Van Emden and Kowalski, [16], or Apt and Van Emden, [1], page 845) but a simple proof is worth mentioning.

**Lemma 4.2** Given a set $H$ of equational ground Horn clauses, given any $S(H)$-indexed family $R$ of relations on $TERM(H)$, there is a smallest implicational relation $\mathcal{D}_R$ on the graph $GT(H)$ containing $R$. The relation $\mathcal{D}_R$ is called the implicational closure of $R$ on $GT(H)$.

**Proof:** We define the sequence $R^i$ of $S(H)$-indexed families of relations inductively as follows:

for every sort $s \in S(H)$, for every $i \geq 0$,

\[
R_s^0 = R_s \quad \cup \quad \{(u, v) \in TERM(H)^2_s \mid u \doteqdot v \in H\},
\]

\[
R_s^{i+1} = R_s^i \quad \cup \quad \{(u, v) \in TERM(H)^2_s \mid \text{there is some clause } u \doteqdot v : - u_1 \doteqdot v_1, \ldots, u_n \doteqdot v_n \text{ in } H, \text{ such that, } u_jR_{w_j}v_j, \text{ for some sort } w_j, 1 \leq j \leq n \}.
\]

Let $(\mathcal{D}_R)_s = \bigcup_{i \geq 0} R_s^i$. As in the previous proof, it is easily shown that $\mathcal{D}_R$ is the implicational closure of $R$. \qed

Since $GT(H)$ is finite, there is a least integer $i$ such that $R^i = R^{i+1}$. Hence, the implicational closure $\mathcal{D}_R$ of $R$ is computable.

Note that $\mathcal{D}_R$ is not necessarily an equivalence relation, but this does not matter because we are going to interleave implicational closure steps, and equational congruence closure steps. Also notice that $\mathcal{D}_R$ is an order-sorted relation.

4.3 Order-sorted Congruence Closure For Horn Clauses

The idea is to interleave steps in which the implicational closure is computed, and steps in which the equational order-sorted congruence closure is computed.
Theorem 4.3: Given a set $H$ of equational ground Horn clauses, given any $S(H)$-indexed family $R$ of relations on $TERM(H)$, there is a smallest order-sorted congruence closure $\rightarrow^*_R$ on the graph $GT(H)$ containing $R$.

Proof: We define the sequence $R^i$ of $S(H)$-indexed families of relations inductively as follows: For every sort $s \in S(H)$, for every $j \geq 0$,

$$
egin{align*}
R_s^0 &= R_s, \\
R_s^{2j+1} &= \supseteq R_s^{2j}, \\
R_s^{2j+2} &= \subseteq R_s^{2j+1}.
\end{align*}
$$

Let $(\rightarrow^*_R)_s = \bigcup_{i \geq 0} R_s^i$. Since the graph $GT(H)$ is finite, there is some integer $i \geq 2$ such that $R^i = R^{i+1}$. If $i = 2j$, since $R_s^{2j+1} = \supseteq R_s^{2j}$ and $j \geq 1$, then $R_s^{2j}$ is an equational order-sorted congruence, and $R_s^{2j+1}$ is an order-sorted congruence on $GT(H)$. If $i = 2j + 1$, since $R_s^{2j+2} = \subseteq R_s^{2j+1}$ and $j \geq 1$, then $R_s^{2j+1}$ is an implicational order-sorted relation, and $R_s^{2j+2}$ is an order-sorted congruence on $GT(H)$. It can also easily be shown by induction that any order-sorted congruence on $GT(H)$ containing $R$ contains every $R^i$. Hence, $\rightarrow^*_R$ is the order-sorted congruence closure of $R$ on $GT(H)$. \hfill \Box

4.4 A more efficient Order-sorted Graph Congruence closure

There is an aspect of the method described above that makes it somehow unnecessarily computationally expensive: it computes the whole family of relations $R_s$. Within a given connected component it is only necessary to compute the $R_t$ for $t$ the top sort of that component since the order-sortedness of the congruence forces the congruences on the lower sorts to hold. If the signature is coherent, as it is in our case, one can definitely drop the subscript from the congruence. Our discussion on top algebras in section 2.9 can be adapted to graph congruences. As a consequence, one can construct a many-sorted congruence on the top sorts which then naturally extends to an order-sorted congruence when taking the rest of the sorts in consideration. For the purpose of our paper however, it is not even necessary to compute this extension, since by virtue of the signature being coherent, one does not need to differentiate between different components of the same congruence.

The methods for computing the order-sorted graph congruence closure in [2] work in a bottom up fashion disregarding the sorts of the terms (it assumes that every term in $H$ is well typed). Hence they yield a graph congruence closure which has the lower sorts property. Thus the algorithm described there computes the order-sorted graph congruence closure in $O(n^2)$. 
5 Conclusion

We have shown that congruence closure extends to finite sets of ground order-sorted equational Horn clauses. We have also proved that congruence closure is sound and complete for showing the unsatisfiability of sets of ground order-sorted equational Horn clauses (for coherent signatures). As a consequence, adapting results from Gallier [2], an unsatisfiability algorithm running in $O(n^2)$ has been obtained. The proof that the method is complete is surprisingly nontrivial and, as a by-product, it yields an interesting characterization of the least order-sorted congruence containing a relation $R$. Elsewhere, we have shown that congruence closure plays a crucial role in generalizing Andrews's method of matings to first-order languages with equality (Gallier, Raatz, Snyder [4]).

We are hoping that the congruence closure algorithm for order-sorted algebras can be used in a similar way for order-sorted logic.

References


