Soft Typing for Ordered Resolution

Harald Ganzinger, Christoph Meyer, Christoph Weidenbach*

Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken, Germany
email: {hg,meyer,weidenb}@mpi-sb.mpg.de

Abstract. We propose a variant of ordered resolution with semantic restrictions based on interpretations which are identified by the given atom ordering and selection function. Techniques for effectively approximating validity (satisfiability) in these interpretations are presented. They are related to methods of soft typing for programming languages. The framework is shown to be strictly more general than certain previously introduced approaches. Implementation of some of our techniques in the SPASS prover has lead to encouraging experimental results.

1 Introduction

Exploiting knowledge about certain models of a theory for improving automated proof search has received quite some attention in the past since the work by Slagle [1967] on semantic resolution. Related methods in the context of semantic tableaux and clause linking have been proposed by Plaisted [1994] and others. The SATCHMO system [Manthey and Bry 1988] represents one of the more well-known attempts at implementing semantic techniques in an actual prover. In these approaches inferences are filtered by semantic considerations in that they are restricted to premises enjoying particular satisfiability properties with respect to a given interpretation. The interpretation usually is a model of some subset of the input. Two problems have to be solved in this context: a suitable interpretation has to be chosen and validity and/or satisfiability of formulas, in particular clauses, has to be decided for that interpretation.

This paper describes two ideas for approaching these problems in the context of clausal, ordered resolution and superposition. We propose to employ the ordering for identifying a particular model for a specific subset of the clauses. These models will be called *partial interpretations* below. Secondly, we propose methods, related to what has been called "soft typing" in the programming language area [Frühwirth, Shapiro, Vardi and Yardeni 1991], to *automatically* infer abstractions of the clauses for approximating validity and satisfiability with respect to partial interpretations. An abstraction might, for instance, approximate the extension of a predicate such that emptiness of the abstracted predicate implies emptiness of the predicate in any model of the given theory.

* This work was supported by the German science foundation program Deduktion.

In the context of refutational theorem proving, our partial interpretations may be viewed as attempts at constructing a counterexample for disproving the given hypothesis. An atom ordering implicitly identifies one model (among all possible models) that is minimal with respect to a certain extension of the ordering to Herbrand interpretations [Bachmair and Ganzinger 1991]. Ordered resolution focuses search to that particular model rather than exploring all counterexample candidates simultaneously. Like in semantic resolution, our approach to exploiting semantic information will be compatible with ordering restrictions for *any* given ordering, and, therefore, will be compatible with ordering-based simplifications such as reduction. By specifying a suitable ordering, a user identifies preferred models and normal forms of expressions alike. Without the use of orderings, simplification by reduction is virtually impossible and deduction with equality becomes hopelessly inefficient. By choosing a particular model from the ordering, our semantic filtering of resolution inferences will be *more restrictive* than semantic resolution.

Similar to the SATCHMO method and to [Plaisted 1994], the partial interpretations that we use for selecting inferences are *dynamic* in that they are constantly modified throughout the theorem proving process by including results from inferences. Unlike SATCHMO in our approach the modifications are generally *not monotone* since clauses that contribute to partial interpretations need not to be Horn. That makes constraint solving less incremental but, on the other hand, we need not necessarily split disjunctions and we do not have to impose any restrictions related to extra variables in the succedent of a clause.

Technically the paper will define a concept of blocking for inferences that is based on truth in certain partial interpretations. We show that the blocking constraints are strictly more restrictive than the ones for semantic resolution. A set of clauses will be called *weakly saturated* if all non-blocked inferences are redundant. *Redundancy* is, as in our previous work [Bachmair and Ganzinger 1994], based on logical implication from smaller clauses and, hence, independent of any model hypothesis. We show that the SATCHMO saturation criterion is a special case of weak saturation for an appropriately defined atom ordering.

In Sec. 8, we describe a method, derived from what is implemented in the SPASS system [Weidenbach, Gaede and Rock 1996], by which truth in partial interpretations can be *effectively approximated.* It is essential to note that the semantic foundation for the method is independent from these specific techniques of abstract interpretation. However, providing decision and unification procedures for the abstract theories is of central importance to any practical implementation of the method. The paper also reports on computer experiments with the SPASS system, and concludes with a comparison to previous work.

Although the theoretical investigations in this paper are mainly formulated for first-order logic without equality, their extension to the equational case and to the superposition calculus is a standard exercise. However, finding appropriate approximations in the equational case is an order of magnitude more complex due to the fact that decidability results about unifiability in non-trivial fragments of first-order logic with equality are extremely hard to obtain.

2 Preliminaries

We adhere to the usual definitions for variables, substitutions, terms, atoms, (positive and negative) literals, multisets, and clauses. Atoms formed from unary predicates are called *monadic*. Where not specified otherwise, letters A and B denote atoms, L denotes literals, C and D denote clauses, and the letter N denotes a set of clauses. Clauses will be written both in sequent notation $\Gamma \rightarrow \Delta$, where Γ (the *antecedent*) and Δ (the *succedent*) are multisets of atoms, and in multiset notation $L_1, \ldots, L_k$ or $L_1 \vee \ldots \vee L_k$, where the L_i are literals. A clause C is called a *Horn clause* if it contains at most one positive literal. An expression is called *ground* if it contains no variables. If ϕ is a formula, by, respectively, $\forall\phi$ and $\exists\phi$ we denote the universal and existential closure of ϕ.

A *(Herbrand) interpretation* I is a set of ground atoms. A ground atom A is called *true* in I if A is in I, and is called *false* in I, otherwise. A negated atom $\neg A$ is true [false] in I if and only if A is false [true] in I. A ground clause is *true* in I if one of its literals is true in I. If an expression (atom, literal, clause) E is true in I we write $I \models E$ and also say that I *satisfies* E. A Herbrand interpretation I is said to satisfy a non-ground clause if it satisfies all its ground instances. It is called a *model* of a set N of clauses if it satisfies all clauses in N. A clause set N is called *consistent* if it has a model, and it is called *inconsistent*, otherwise.

3 Ordered Resolution

Ordered Resolution is parameterized by orderings on atoms and by selection functions. We briefly review the calculus and the main completeness results for the ground case. Lifting of resolution is straightforward theoretically, although in practice there are a number of choices one can make regarding the approximation of the lifted ordering and equality constraints.

An *atom ordering* is a well-founded, total ordering on ground atoms. Given an atom ordering $\succ$, we will call an atom A [strictly] maximal with respect to a multiset of atoms Γ, if for any B in Γ we have $B \not\succ A$ $[B \not\succeq A]$. Any atom ordering $\succ$ is extended to an ordering on literals by taking the multiset extension of $\succ$ and by identifying any positive literal A with the singleton $\{A\}$ and any negative literal $\neg A$ with the multiset $\{A, A\}$. With this definition, $\neg A$ is greater than A, but is smaller than any literal B or $\neg B$ with $B \succ A$. The multiset extension of the literal ordering induces an ordering on ground clauses. Let us also use $\succ$ to denote both the extension to literals and clauses of any given atom ordering $\succ$. The clause ordering is compatible with the atom ordering; if the maximal atom in C is greater than the maximal atom in D then $C \succ D$. If N is a set of ground clauses and C a ground clause (not necessarily in N), N_C denotes the set of clauses D in N such that $C \succ D$. We say that a clause $C \vee A$ is *reductive for* the atom A, if A is a strictly maximal atom with respect to C.

A *selection function* S assigns to each ground clause a possibly empty set of occurrences of negative literals. If C is a clause, the literal occurrences in $S(C)$ are called *selected*. $S(C) = \emptyset$ indicates that no literal is selected.

Let $\succ$ be an atom ordering on ground atoms and let S be a selection function. An inference by *ordered resolution (with selection)* between ground clauses takes the form

$$\frac{C \vee A \qquad \neg A \vee D}{C \vee D}$$

such that (i) $C \vee A$ is reductive for A, (ii) no literal is selected in C, and (iii) $\neg A$ either is selected, or else is maximal with respect to D. We call $C \vee A$ and $\neg A \vee D$ the *positive* and the *negative* premise, respectively, of the inference and $C \vee D$ the conclusion.

An inference by *ordered factoring* takes the form

$$\frac{C \vee A \vee A}{C \vee A}$$

such that (i) A is maximal with respect to C, and (ii) no literal is selected in the premise.

Note that inferences by ordered resolution in which the first premise contains a selected literal are excluded, as are inferences by ordered factoring from clauses with selected literals.

The notion of redundancy to be introduced next is deemed to identify clauses and inferences which, due to the presence of other clauses and inferences in a particular theorem proving context, are not needed for obtaining a contradiction. Redundancy is defined with respect to the given ordering. Let N be a set of ground clauses. A ground clause C (not necessarily a member of N) is called *redundant in* N if it is entailed by the members of N which are smaller than C, i.e., if $N_C \models C$.

Let $C' \vee D'$ be the conclusion of an inference by ordered resolution from ground clauses $C' \vee A$ and $D = \neg A \vee D'$. We call the inference *redundant in* N if $N_D \models C' \vee D'$, that is, the conclusion is entailed by clauses in N smaller than the negative premise. An inference by ordered factoring is called *redundant in* N if the conclusion follows from clauses in N which are smaller with respect to $\succ$ than its premise. Finally, we call a set N of ground clauses *saturated up to redundancy* (with respect to $\succ$ and a selection function S) if any inference by ordered resolution and factoring from non-redundant premises in N is redundant in N.

4 Semantic Foundations

Partial interpretations. Semantic filtering of inferences may be based on interpretations I that are derived from the given set of clauses and from the ordering. The interpretation I serves as a model hypothesis and is constructed from a subset of clauses which, due to reductivity properties, is necessarily satisfiable. If all clauses are true in I then no further inference is required. Otherwise, resolution inferences only need to consider, for the negative premises, clauses that are false in I, and, for the positive premises, clauses that effectively contribute to I. Ideally, the newly inferred clauses lead directly to an appropriate modification of the model hypothesis. The model construction that we have used for obtaining completeness proofs of various calculi for resolution, chaining, and paramodulation [Bachmair and Ganzinger 1994] will be helpful in this regard.

Let $\succ$ be a total atom ordering and S a selection function. Given a set of ground clauses N, we use induction with respect to $\succ$ to define a Herbrand interpretation I_C and a set ε_C, for each clause C in N, as follows.

Definition 1. Let I_C be the set $\bigcup_{C \succ D} \varepsilon_D$. Furthermore, $\varepsilon_C = \{A\}$ if (i) $C = C' \vee A$ is reductive for A, (ii) C contains no selected atom, and (iii) C is false in I_C. Otherwise, ε_C is the empty set.

If $\varepsilon_C = \{A\}$, we also say that C *produces* A and call C a *productive clause*. Finally, by I, we denote the Herbrand interpretation $\bigcup_{C \in N} \varepsilon_C$. Whenever we need to emphasize the dependency of the interpretations I_C and I from the particular clause set N, we will use the notations I_C^N and I^N, respectively. Moreover, for any clause C the interpretations I_C^N, I^{N_C} and $\bigcup_{D \in N_C} \varepsilon_D$ coincide.

The construction is designed to render the formulas of N true in I^N. The interpretation I_C, called the *partial interpretation up to* C, is intended to be a model of the set N_C of those clauses in N that are smaller than C. The interpretation ε_C is meant to be a minimal extension of I_C that makes C true. However, if N is not saturated I^N will only satisfy a subset of the clauses in N, cf. Theorem 2 below.

Blocking. A partial interpretation I can be viewed as representing a model hypothesis for N that is based on the currently available knowledge about N. If a clause C falsifies the hypothesis, more inferences are required. Conversely, clauses which have the appropriate truth value need not be considered for inferences. This idea is the basis for our notion of blocked inferences.

Let N be a set of ground clauses. We say that an inference by ordered resolution with positive premise C and negative premise D is *blocked in* N, if (i) C is false in I^N, or (ii) C is true in I_C^N, or (iii) D is true in I^N. Basically, inferences by resolution can be restricted to cases in which the positive premise is a productive clause and the negative premise is false. An inference by ordered factoring from C is called *blocked in* N if C is true in I^N. Note that, due to the inductive and monotone construction of I, a clause C is true in I if and only if the clause is true in the partial interpretation $I_C \cup \varepsilon_C$, and that ε_C is non-empty if and only if C is productive. A clause set N is called *weakly saturated* if any non-blocked inference from non-redundant clauses in N is redundant in N. The blocking criterion (ii) for resolution inferences represents an additional semantic filter compared to semantic resolution, where, if we abstract from the polarity of literals, only (i) and (iii) apply [Slagle 1967].

Example. For an example, assume the atom ordering $B \succ A$ and an empty selection function. The following table describes the construction of I for an inconsistent set of clauses which the table lists in ascending order.

Clause C	I_C	ε_C	Remarks
$\rightarrow A, B$	$\emptyset$	$\{B\}$	false in I_C, productive, B is maximal
$A \rightarrow B$	$\{B\}$	$\emptyset$	true in I_C
$B \rightarrow A$	$\{B\}$	$\emptyset$	false in I_C, B is maximal
$A, B \rightarrow$	$\{B\}$	$\emptyset$	true in $I_C = I$

According to our definition, the only non-blocked inference is by resolution from $\rightarrow A, B$ and $B \rightarrow A$, yielding (after factoring) the unit clause $\rightarrow A$. Constructing the partial interpretations for the new clause set proceeds as follows:

Clause C	I_C	ε_C	Remarks
$\rightarrow A$	$\emptyset$	$\{A\}$	false in I_C, productive, A is maximal
$\rightarrow A, B$	$\{A\}$	$\emptyset$	true in I_C
$A \rightarrow B$	$\{A\}$	$\{B\}$	false in I_C, productive, B is maximal
$B \rightarrow A$	$\{A, B\}$	$\emptyset$	true in I_C
$A, B \rightarrow$	$\{A, B\}$	$\emptyset$	now false in $I_C = I$, B is maximal

Again, only one inference is non-blocked, the resolution inference from the third and last clause, respectively. From this we derive $A \rightarrow$, and then, by reduction (or a resolution step) with the first clause, the contradiction. We observe that by choosing counterexamples (false clauses in I) according to any well-founded ordering, computing non-blocked inferences can be made a completely deterministic refutation process, whereas ordering restrictions alone or semantic resolution is not deterministic.

Refutational Completeness.

Theorem 2. *If N is weakly saturated and contains no contradiction, then (i) I is a model of N, (ii) for any ground clause C, I_C is a model of N_C, and (iii) for any clause C in N, $I_C \cup \varepsilon_C$ is a model of $N_C \cup \{C\}$.*

Proof. The proof is a direct consequence of the Lemmas 4.12, 5.3, and 5.5 in [Bachmair and Ganzinger 1994]. What is called "blocked" here has been called "redundant" in the cited paper. What we call "redundant" now, has been called "composite" in [Bachmair and Ganzinger 1994].

As a corollary one obtains from (i) that weakly saturated sets N are either consistent or contain the empty clause. In [Bachmair and Ganzinger 1991] the above statement (i) is strengthened in that I is shown to be the unique minimal ("perfect") model of N. Here, the ordering $\succ^i$ on interpretations is the multiset extension of the *inverse* $\prec$ of the atom ordering $\succ$. Note that the existence of minimal models (with respect to $\succ^i$) for consistent, infinite sets of clauses is not trivial, as $\succ^i$ is not well-founded in general.

Horn Clauses. Of particular interest is the case of Horn clauses. For satisfiable sets H of Horn clauses it turns out that one may always find an ordering for which H is weakly saturated and for which the construction I yields the minimal model of H. The ordering has to be constructed from the T_P-operator known from logic programming.

Let H be a set of Horn clauses. The function T_H maps interpretations J to interpretations $T_H(J)$ by

$$T_H(J) = \{A \mid \exists A_1, \ldots, A_n \to A \in H : \ A_i \in J, \text{ for } 1 \le i \le n\} \cup J.$$

It is well-known that $J^H = \bigcup_{n \ge 0} T_H^n(\emptyset)$ is the minimal model of H. Let ι_A, for any ground atom A in J^H, denote the minimal index m for which A is in $T_H^m(\emptyset)$. For atoms A not in J^H we set $\iota_A = \infty$. If $\succ$ is an atom ordering such that $A \succ B$ whenever $\iota_A > \iota_B$, then we call $\succ$ *compatible* with H. For compatible atom orderings the construction I from Definition 1 is simply another method for generating the minimal model of H. More precisely we have:

Proposition 3. *Let H be a satisfiable subset of Horn clauses of some clause set N, and let $\succ$ be compatible with H. Let C be a ground clause in N, let $(\neg)A$ be maximal in C and let B be some ground atom.*

1. *If $A \succ B$ then B is in J^H if and only if B is in I_C^H.*
2. *If $\neg A$ in C or $\{A, A\} \subseteq C$ then A is in J^H if and only if A is in I_C^H.*
3. *H is weakly saturated with respect to $\succ$.*
4. *$I_C^H \subseteq I_C^N$.*

Proof. We show (4) by induction on C and leave (1)–(3) to the reader. Suppose that A is in I_C^H. Then there exists a clause $D = \Gamma \to A$ in H_C which produces A into I^H. In particular, any B in Γ is smaller than A and $\Gamma \subseteq I_D^H$. Using the induction hypothesis for D we may infer that $\Gamma \subseteq I_D^N$. Therefore, either A is in I_D^N, or else D produces A into I^N. In both cases we conclude that A is in I_C^N.

5 Effective Saturation Strategies

When we speak of *deduction* in the sequel we mean the derivation of any sound consequence, possibly by, but not restricted to, ordered resolution and factoring. Non-ordered inferences, although not strictly required for refutational completeness, might be useful for simplification. For example, from clauses $\neg A$ and $A \vee B$, with $B \succ A$, we may infer by a non-ordered step of resolution the clause B which, when added to the current set of clauses, would cause $A \vee B$ to be redundant.

A (finite or countably infinite) sequence N_0, N_1, N_2, ... of sets of ground clauses is called a *theorem proving derivation* if each set N_{i+1} can be obtained from its predecessor N_i either (i) by adding a set of clauses that can be deduced from N_i, or else (ii) by deleting a subset of clauses which are all redundant in N_i. The set $N_\infty = \bigcup_j \bigcap_{k \ge j} N_k$ is called the *limit* of the derivation. A theorem proving derivation is called *fair* if every non-blocked inference by ordered resolution or ordered factoring from premises in N_∞ is redundant with respect to $\bigcup_j N_j$.

Lemma 4. *The limit N_∞ of a fair theorem proving derivation is weakly saturated, and the clauses in $(\bigcup_j N_j) \setminus N_\infty$ are redundant in N_∞.*

Proof. First, if N is a subset of some N' such that clauses in $N' \setminus N$ are redundant with respect to N', then any clause or inference that is redundant in N' is also redundant in N. Second, if the sequence of N_i is a theorem proving derivation, any clause C in $(\bigcup_j N_j) \setminus N_\infty$ is redundant in some N_j, hence redundant in $\bigcup_j N_j$. If the theorem derivation is fair, every non-blocked inference from N_∞ is redundant in $\bigcup_j N_j$ and, therefore, using the first statement, redundant in N_∞.

Theorem 5. *Let N_0, N_1, N_2, ... be a fair theorem proving derivation. If $\bigcup_j N_j$ does not contain the empty clause, then N_∞ is weakly saturated and N_0 is consistent.*

It is not difficult to generalize the above notion of linear theorem proving derivations to theorem proving *derivation trees* by admitting deduction steps to split a set of clauses N into $k \geq 1$ alternatives $N \cup M_1, \ldots, N \cup M_k$ of clause sets $N \cup M_i$ such that N is consistent if and only $N \cup M_i$ is consistent for some $1 \leq i \leq k$. For example, we might split $N \cup \{C \vee D\}$ on a clause $C \vee D$ with variable-disjoint subclauses C and D into two branches $N \cup \{C\}$ and $N \cup \{D\}$. A derivation tree is *fair* if each path is fair. For a fair tree, the limit system N^π_∞ of each (possibly infinite) path π in the tree is weakly saturated,[1] and N_0 is inconsistent if and only if on every path π the empty clause is in $\bigcup_j N^\pi_j$. SPASS computes tree-like derivations with splitting on variable-disjoint clause parts.

6 SATCHMO

We demonstrate that the semantic methods in SATCHMO [Manthey and Bry 1988] are an instance of our concept. The SATCHMO theorem prover, given a set N of clauses, computes consistent subsets H of Horn clauses of N and then selects clauses which are false in the minimal model of H in order to compute certain hyper-resolution inferences from them. If all clauses are true in the minimal model of H then the set is considered saturated, as it is, obviously, consistent. We show that this particular saturation criterion, for any set H of ground Horn clauses, is a special case of our notion of weak saturation. Throughout this section, $\succ$ will denote any atom ordering that is compatible with H.

Let C be a possibly non-Horn ground clause with maximal atom A. By K^H_C we denote the interpretation $I^H_C \cup \{A\}$ whenever A is in the minimal model J^H and C is reductive for A. Otherwise, $K^H_C = I^H_C$. That is, K^H_C is different from I^H_C only if C is reductive for A and if A gets eventually produced into J^H by a clause D in H such that $D \succeq C$.

Theorem 6. *Let N be a set of ground clauses and H a consistent Horn subset of N. Suppose that J^H satisfies N.*
(1) For any ground clause C, we have $I^N_C \subseteq K^H_C$.
(2) Any non-reductive clause C in N is true in I^N_C.

[1] Note that the Lemma 4 is true regardless of the soundness of deduction steps and can, therefore, be applied to any path in the deduction tree.

Proof. We prove (1) by induction on C. Suppose that A is in I_C^N. Then there exists a clause $D = \Gamma \to A, \Delta$ in N_C which produces A into I^N. In particular, any B in Γ, Δ is smaller than A, and $\Gamma \subseteq I_D^N$, as well as $\Delta \cap I_D^N = \emptyset$. Using the induction hypothesis for D we may infer that $\Gamma \subseteq K_D^H$, hence, by Proposition 3 we get $\Gamma \subseteq J^H$. By the same proposition we conclude $\Delta \cap J^H = \emptyset$. For J^H to satisfy D, the atom A must therefore be true in J^H. By case analysis of whether A occurs in C, we conclude that A is in K_C^H.

Part (2) is an immediate consequence of (1) by observing that, according to (1) and Proposition 3, the interpretations I_C^N and K_C^H coincide for non-reductive clauses C and assign the same truth values as J^H to the atoms in C.

Corollary 7. *Let N be a set of ground clauses and let H be a consistent Horn subset of N. If J^H satisfies N then N is weakly saturated with respect to any ordering that is compatible with H and any selection function.*

This corollary indicates that the SATCHMO saturation criterion is a special case of weak saturation. SATCHMO theorem proving processes are theorem proving derivation trees. By hyper-resolving on false clauses, new positive ground[2] clauses are obtained and the process branches with regard to the disjuncts.

7 Abstraction

Theorem 5 provides a general framework for deferring inferences due to blocking. In the context of ground clauses this framework is effective. Given an (effective) atom ordering $\succ$ and selection function, we can decide whether a clause C is valid in the partial model I_C^N, whenever C is a ground clause and N is a set of ground clauses that is "nicely" represented. For general first-order clause sets N and clauses C validity (satisfiability) is, of course, not decidable. To make effective use of Theorem 5, validity and/or satisfiability have to be safely approximated. In general, an approximation of an interpretation I is a class of interpretations $\mathcal{J}$ such that for a certain class $\mathcal{F}$ of formulas, (i) either validity (satisfiability) in $\mathcal{J}$ implies validity (satisfiability) in I, or, conversely, validity (satisfiability) in I implies validity (satisfiability) in $\mathcal{J}$, and, moreover, (ii) validity (satisfiability) in $\mathcal{J}$ is decidable for the formulas in $\mathcal{F}$.

In the subsequent section this particular form of approximation will be used: An interpretation J is called an *upper approximation* of another interpretation I, if $I \subseteq J$. Suppose that J is an interpretation in which solvability of conjunctions of atoms is decidable. Consider a general clause $C = B_1, \ldots, B_n \to \Delta$ and assume that J is an upper approximation of any I_D, for the ground instances D of C. If $J \not\models \exists\,(B_1 \wedge \ldots \wedge B_n)$ then $I_{C\sigma} \not\models B_1\sigma \wedge \ldots \wedge B_n\sigma$, for any ground instance $C\sigma$ of C. In this case, any inference with premise C is blocked.

Similarly, an interpretation J is called a *lower approximation* of another interpretation I, if $J \subseteq I$. If $C = \Gamma \to A, A_1, \ldots, A_n$ and if $J \models \forall\,(A_1 \vee \ldots \vee A_n)$,

[2] Groundness is guaranteed by imposing a restriction on clauses by which extra variables in the succedent are not admitted.

then C cannot be the positive premise of a non-blocked inference on A. In fact, this would imply that all instances D of C are true in I_D.

8 Sort Theories in Spass

Spass is an automated theorem prover for first-order logic with equality [Weidenbach et al. 1996]. It is an implementation of superposition with semantic blocking of inferences based on a powerful sort logic. Certain conjunctions of monadic atoms in the antecedent of a clause are dealt with specifically and are called *sort constraints*. We write constrained clauses as

$$\Theta \,\|\, \Lambda \to \Pi$$

where Θ is the sort constraint. Constraint atoms in initial clauses have to be of the form $S(x)$ with S a sort (monadic) predicate and x a variable. A sort constraint Θ in a clause $\Theta \,\|\, \Lambda \to \Pi$ is called *solved*, if $vars(\Theta) \subseteq vars(\Lambda \cup \Pi)$ and all terms occurring in Θ are variables. To clauses with solved constraints the usual superposition inference rules are applied, except that sort constraints are not resolved or superposed. Unsolved sort constraints are selected, and specific versions of resolution are applied to transform the sort constraint into solved form. These inference rules implement sorted unification [Weidenbach 1996] on the sort constraint literals. The results throughout this section apply to arbitrary atom orderings and to selection functions which select some negative atom in any clause that has a solved sort constraint. Furthermore, the notion of a blocked sort constraint we will introduce in the sequel, is compatible with equality reasoning in a superposition framework with dynamic sort theories, as the sort theory will be constantly revised according to derived equalities.

Dynamic Sort Theories. The blocking concept developed in Sec. 4 and 7 can be instantiated for an approximation of sort constraints by dynamic sort theories.

Definition 8. The *dynamic sort theory* of a set N of clauses is the set of Horn clauses $\mathcal{L}^+$ consisting of all the clauses $\Theta \,\|\, \to S(t)$ for which there exists a clause $C = \Psi \,\|\, \Lambda \to \Delta, S(t)$ in N such that (i) Ψ is solved, (ii) Θ is a maximal subset of Ψ with $vars(\Theta) \subseteq vars(S(t))$, and (iii) $S(t)$ is strictly maximal with respect to C.

Observe that the sort theory $\mathcal{L}^+$ abstracts from all antecedent literals of clauses in N. In what follows we call a sort constraint $A_1 \wedge \ldots \wedge A_k$ *false* in a set H of Horn clauses whenever $H \not\models \exists\,(A_1 \wedge \ldots \wedge A_k)$, which is the same as saying that $\exists\,(A_1 \wedge \ldots \wedge A_k)$ is false in the minimal model of H. Let N be a set of clauses. The sort constraint of a clause $\Psi \,\|\, \Lambda \to \Delta$ is called *blocked* in N, if Ψ is false in $\mathcal{L}^+$. An inference is *blocked* if either the sort constraint of a premise or the sort constraint of the conclusion is blocked.

Lemma 9. *Let M be the set of all ground instances from clauses in N. If the sort constraint Ψ of a clause $C = \Psi \,\|\, \Lambda \to \Delta$ is blocked in N, then, for any ground instance $C\sigma$ of C, $C\sigma$ is true in $I^M_{C\sigma}$.*

Proof. By contradiction. Let $C\sigma$ be a ground instance of C and assume that $C\sigma$ is false in $I^M_{C\sigma}$. Moreover let J denote the minimal model of $\mathcal{L}^+$. We show by induction that for any ground instance $\Theta\tau \parallel \Gamma\tau \to \Pi\tau, S(t)\tau$ of a clause in N that produces a monadic atom $S(t)\tau$ into $I^M_{C\sigma}$, the atom $S(t)\tau$ is contained in J. A productive clause has no selected atom, hence the constraint Θ must be solved. Then there exists a clause $\Theta' \parallel \to S(t)$ in $\mathcal{L}^+$ such that, by induction hypothesis, $\Theta'\tau \subseteq \Theta\tau$, hence $S(t)\tau$ is in fact in J. The minimal model J of $\mathcal{L}^+$ is an upper approximation for $I^M_{C\sigma}$ with respect to ground sort atoms. Therefore, if $C\sigma$ is false in $I^M_{C\sigma}$ then $\Psi\sigma \subseteq I^M_{C\sigma}$, and hence $\Psi\sigma \subseteq J$, which contradicts the assumption that Ψ is blocked in N.

Theorem 10. *If C is a clause with a sort constraint that is blocked in N then any inference from C is blocked in N.*

Proof. By Lemma 9 and Theorem 5.

Note that in order to exploit the theorem in a fair theorem proving strategy, the abstractions $\mathcal{L}^+$ have to be recomputed whenever a potentially productive clause is added. Our sort theories are *dynamic*. The more logical consequences are being added during the proving process, the better the sort theory abstraction approximates the extension of the sort predicates in the perfect model of a consistent theory.

In order to effectively block the sort constraint of a clause we must be able to decide falsity of constraints in $\mathcal{L}^+$. Even for the very restricted monadic Horn theories $\mathcal{L}^+$ the problem remains undecidable in general. However, we can further abstract from $\mathcal{L}^+$ by removing certain (not necessarily all!) non-linear variable occurrences in the heads of the clauses through introducing fresh variables. Different choices for such a renaming lead to different classes of decidable theories. The problem of solvability in these theories is the problem of sorted unification. Sorted unification has been extensively studied and many decidability and undecidability results, even for sort theories allowing for non-linear variable occurrences, are known [Weidenbach 1996].

Example. Let us consider a simple example where $\succ$ is an atom ordering induced by the ordering $R \succ S \succ T \succ Q$ on the predicate symbols. Let N consist of these clauses, where we mark maximal atoms with *:

$$
\begin{array}{lrll}
(1) & \parallel & & \to Q(a)^* \\
(2) & \parallel & & \to R(a,a)^* \\
(3) & \parallel & & \to S(a)^*, T(a) \\
(4) & Q(x), T(x) \parallel & R(x,x)^* & \to S(x) \\
(5) & S(x) \parallel & R(x,y)^* & \to T(y) \\
(6) & T(y) \parallel & & \to R(x, f(f(y)))^* \\
(7) & T(x) \parallel & R(x, f^4(x))^* & \to
\end{array}
$$

The dynamic sort theory $\mathcal{L}^+$ with respect to N consists of the two facts $Q(a)$ and $S(a)$ generated from the clauses (1) and (3), respectively. The sort constraints of clauses (4), (6), and (7) are blocked, because T is empty with respect to $\mathcal{L}^+$. There is only one possible resolution step between (2) and (5) yielding (after solving the sort constraint with (3))

$$(8) \qquad \| \qquad \rightarrow T(a)^*$$

Now clause (3) becomes redundant and $\mathcal{L}^+$ changes to $\{Q(a), T(a)\}$. The clauses (4), (6) and (7) are no longer blocked. Nevertheless, the resolution inferences between (4), (6) and and (6), (7) result in clauses with a blocked sort constraint and are therefore blocked. The only possible inference is between (2) and (4) eventually generating

$$(9) \qquad \| \qquad \rightarrow S(a)^*$$

The atom $S(a)$ is added to $\mathcal{L}^+$ and now the inference between (5) and (6) is no longer blocked, resulting in

$$(10) \qquad T(x) \,\| \qquad \rightarrow T(f(f(x)))^*$$

The clause (10) is added to $\mathcal{L}^+$ and a final resolution step between (6) and (7) results in the empty clause.

Static Sort Theories. The present implementation of SPASS computes *static sort theories* $\mathcal{L}^*$ for the initial clause set N which, due to a more crude method of abstraction, can be shown to safely approximate *any* of the dynamic theories for any theorem proving process for N. The approximation $\mathcal{L}^*$ consists of the clauses $\Theta \,\| \rightarrow S(t)$ for which there exists a clause $\Psi \,\| \,\Lambda \rightarrow \Delta, S(t)$ in N such that (i) Ψ is solved and (ii) Θ is a maximal subset of Ψ with $vars(\Theta) \subseteq vars(S(t))$. The definition of $\mathcal{L}^*$ differs from $\mathcal{L}^+$ in that the requirement about the maximality of $S(t)$ has been dropped. Clearly, the minimal model of $\mathcal{L}^*$ is an upper approximation of the minimal model of $\mathcal{L}^+$. Moreover, $\mathcal{L}^*$ is stable under the addition of logical consequences and under the deletion of redundant clauses.

Hence, clauses that have a false sort constraint in the sort theory $\mathcal{L}^*$ of the initial clauses can be deleted as all inferences from them will be blocked forever.

With respect to our above example, the static sort theory $\mathcal{L}^*$ contains the ground facts $Q(a)$, $S(a)$, $T(a)$ and the two clauses $Q(x), T(x) \,\| \rightarrow S(x)$, $\| \rightarrow T(y)$. The sort T collapses to include arbitrary elements. Nevertheless, the possible inference between (4) and (6) results in a sort constraint that is unsolvable with respect to $\mathcal{L}^*$ allowing to delete the conclusion of this inference.

Unfortunately, the concept of static sort theories cannot easily be extended to equality handling by superposition, as the required approximation properties do not hold for equality interpretations. An extension to this case requires the incorporation of (abstracted) equations to sort theories. Decidability results for such theories that still allow for a rich term structure are extremely hard to obtain.

Experiments. In order to provide experimental evidence about the effects of blocking based on the static sort theory $\mathcal{L}^*$, we have run SPASS on several hundred knowledge bases written in $\mathcal{ALC}$. The language $\mathcal{ALC}$ is a notational variant of multi-modal propositional logic K that can be translated into first-order logic such that resolution with subsumption and condensing becomes a decision procedure for the resulting fragment of first-order clause logic [Schmidt 1997]. Hence any resolution based prover with subsumption and condensing can be used as a decision procedure for this class. The deletion of clauses that have a false sort constraint with respect to $\mathcal{L}^*$ is one of the key techniques to make resolution an "efficient" decision procedure on these problems. The table shows the names of several, representative problems[3] the number of derived clauses, kept clauses[4] and the number of clauses deleted as a result of a false sort constraint in $\mathcal{L}^*$. These clauses cannot be deleted by usual redundancy criteria. Although the problems are fairly easy for SPASS (SPASS solves all these problems in less than 10 seconds on a Sun Sparc Ultra 170/E), without the blocked constraint deletion rule they become significantly harder (at least a factor of 2, except for the "easy" final two examples). Furthermore, these problems are not easy for provers that do not have a sort concept at all. The reader is encouraged to try the problems on any (complete) prover of her/his choice.

Problem	Status	Derived	Kept	Deleted using $\mathcal{L}^*$
5-1-154-3-2	satisfiable	929	233	7
4-7-2-2-0	satisfiable	1156	400	369
4-9-2-2-0	unsatisfiable	139	61	37
8-1-10-4-2	satisfiable	867	276	349
8-1-8-4-2	satisfiable	111	49	42
5-2-80-3-2	unsatisfiable	32	12	0
5-1-85-3-2	unsatisfiable	15	15	0

9 Related Work

Due to space limitations we can only discuss some relationships to the literature: The semantic prerequisites described in Sec. 4 have been developed in our earlier papers [Bachmair and Ganzinger 1994] and [Bachmair and Ganzinger 1991] but have not been exploited before for developing semantics-based calculi for resolution or paramodulation. This is partly due to the fact that all the commonly proposed techniques for simplification and deletion can already be justified by the model-independent notion of redundancy.

Semantic resolution [Slagle 1967] assumes some user-given interpretation I and filters inferences by hyper-resolution with respect to truth in I. Semantic resolution is compatible with ordering constraints for the electrons, but not for

[3] All problems are contained in the SPASS distribution which can be freely obtained from http://www.mpi-sb.mpg.de/guide/software/spass.html.

[4] Clauses backtracked due to applications of the splitting rule of SPASS are not considered to be kept clauses.

the nucleus, in the inferences. Our approach improves semantic resolution in that the semantic restrictions for inferences are stronger. On the other hand, to actually exploit this theoretical improvement requires to dynamically recompute the (approximations of the) partial interpretations such as with our dynamic sort theories. Moreover, by choosing I, a user may provide additional semantic intuition where our automatically selected interpretations are syntactic.

Plaisted [1994] describes a semantic variant of hyperlinking based on essentially the same concept of partial interpretations as ours. As a consequence, his method will also yield a deterministic refutation process on the ground level. The author also stresses the importance of including user-given semantic information and considers those for his construction of partial interpretations. Our constructions can be extended in a similar way. Little is said, however, about how to *automatically* construct models for first-order clauses and how to *automatically* approximate validity and solvability such that they become decidable.

The SATCHMO prover [Manthey and Bry 1988] implements an instance of our framework as we have shown in Sec. 6.

10 Conclusions and Further Work

We have presented a concept of semantic ordered resolution in which models are implicitly specified by the ordering. Our approach is especially suitable for applications in which semantics-based proof search, for whatever reason, should be fully automatic. Ordered versions of resolution are sometimes criticized on the grounds that orderings have to be provided as an additional input. Our own experience shows that good orderings can often be derived automatically by an analysis of the recursion structure in predicate definitions. Furthermore, even if we (automatically) choose naive instantiations for the ordering, restricting inferences by an ordering is still often more convenient than applying no ordering restrictions at all.

Our notion of blocking for inferences is compatible with the usual simplification and deletion techniques. We have shown that our semantic theory is sufficiently general to study various methods including semantic resolution, the SATCHMO approach, and the sort theory-based techniques in SPASS. We have advocated to apply methods related to soft typing for programs in order to approximate computation in the inferred partial interpretations. We have described two instances of such approximation, the dynamic and static sort theories, the latter of which has been implemented in the SPASS prover. We have presented examples where these methods lead to much smaller search spaces.

Future work will, among others, aim at obtaining experimental evidence for the usefulness of dynamic sort theories. These have the advantage of being compatible with equality reasoning. Then we would like to identify further decidable fragments of first-order logic with equality that would allow us to extend our ideas effectively to equational constraints or static approximations including equality. Decidability results about unification in theories saturated by basic paramodulation [Nieuwenhuis 1996] or about approximating reachability in term rewriting systems by using tree automata [Comon 1995, Jacquemard 1996] may turn out to be helpful in this respect.

References

Bachmair, L. and Ganzinger, H. [1991], Perfect model semantics for logic programs with equality, *in* 'Proc. International Conference on Logic Programming '91', MIT Press, pp. 645–659.

Bachmair, L. and Ganzinger, H. [1994], 'Rewrite-based equational theorem proving with selection and simplification', *Journal of Logic and Computation* **4**(3), 217–247. Revised version of Max-Planck-Institut für Informatik technical report, MPI-I-91-208, 1991.

Comon, H. [1995], Sequentiality, second order monadic logic and tree automata, *in* 'Proceedings 10th IEEE Symposium on Logic in Computer Science, LICS'95', IEEE Computer Society Press, pp. 508–517.

Frühwirth, T., Shapiro, E., Vardi, M. Y. and Yardeni, E. [1991], Logic programs as types for logic programs, *in* A. R. Meyer, ed., 'Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science, LICS'91', IEEE Computer Society Press, pp. 300–309.

Jacquemard, F. [1996], Decidable approximations of term rewriting systems, *in* H. Ganzinger, ed., 'Rewriting Techniques and Applications, 7th International Conference, RTA-96', Vol. 1103 of *LNCS*, Springer, pp. 362–376.

Manthey, R. and Bry, F. [1988], Satchmo: A theorem prover implemented in prolog, *in* E. Lusk and R. Overbeek, eds, '9th International Conference on Automated Deduction,CADE-9', Vol. 310 of *LNCS*, Springer, pp. 415–434.

Nieuwenhuis, R. [1996], Basic paramodulation and decidable theories (extended abstract), *in* 'Proceedings 11th IEEE Symposium on Logic in Computer Science, LICS'96', IEEE Computer Society Press, pp. 473–482.

Plaisted, D. A. [1994], Ordered semantic hyper-linking, Technical report, Max-Planck-Institut für Informatik, Saarbrücken.

Schmidt, R. A. [1997], Resolution is a decision procedure for many propositional modal logics, Technical Report MPI-I-97-2-002, Max-Planck-Institut für Informatik, Saarbrücken, Germany. The extended abstract version is to appear in Kracht, M., de Rijke, M., Wansing, H. and Zakharyaschev, M. (eds) (1997), *Advances in Modal Logic'96*, CSLI Publications, Stanford.

Slagle, J. R. [1967], 'Automatic theorem-proving with renamable and semantic resolution', *Journal of the ACM* **14**, 687–697.

Weidenbach, C. [1996], Computational Aspects of a First-Order Logic with Sorts, Dissertation, Technische Fakultät der Universität des Saarlandes, Saarbrücken, Germany.

Weidenbach, C., Gaede, B. and Rock, G. [1996], SPASS & FLOTTER, Version 0.42, *in* M. McRobbie and J. Slaney, eds, '13th International Conference on Automated Deduction, CADE-13', Vol. 1104 of *LNAI*, Springer, pp. 141–145.