Embedding Deduction Modulo into a Prover

Guillaume Burel

Max Planck Institute for Informatics
Saarland University
Saarbrücken, Germany
guillaume.burel@ens-lyon.org http://www.mpi-inf.mpg.de/~burel/

Abstract. Deduction modulo consists in presenting a theory through rewrite rules to support automatic and interactive proof search. It induces proof search methods based on narrowing, such as the polarized resolution modulo. We show how to combine this method with more traditional ordering restrictions. Interestingly, no compatibility between the rewriting and the ordering is requested to ensure completeness. We also show that some simplification rules, such as strict subsumption eliminations and demodulations, preserve completeness. For this purpose, we use a new framework based on a proof ordering. These results show that polarized resolution modulo can be integrated into existing provers, where these restrictions and simplifications are present. We also discuss how this integration can actually be done by diverting the main algorithm of state-of-the-art provers.

Whatever their applications, proofs are rarely searched for without context: mathematical proofs rely on set theory, or Euclidean geometry, or arithmetic, etc.; proofs of program correctness are done using e.g. pointer arithmetic and/or theories defining data structures (chained lists, trees, ...); concerning security, theories are used for instance to model properties of encryption algorithms. It is therefore essential to have theoretical foundations and practical methods that handle theories conveniently and efficiently. For this purpose, there are two directions: to develop methods that are really specific to a particular theory; or to develop a generic framework that can handle all theories. The first option is appealing for efficiency reasons: for instance, combining a SAT solver with the Simplex method leads to very powerful SMT solvers for linear arithmetic. However, developing methods for new theories is hard. Even the combination of such specific methods is not trivial, although there have been a lot of interesting results in that direction in the recent years. In this paper, we are more interested in the second option: having a generic way to handle theories efficiently. A naive way to do so would be to use an axiomatization of the theory, but in general, this approach would be really inefficient for automated proving. Somehow, we need to present the theory so as to take advantage of its properties.

A first idea is to use the consistency of the theory. When proving a goal in a consistent theory by refutation, resolving the clauses of the theory is useless, since it will not bring out a contradiction. This idea defines the set-of-support

strategy for resolution [1], where clauses generated by resolution must have at least one parent outside the theory. The completeness of this method can be proved provided the theory is consistent. However, unless the theory is saturated, this strategy is not compatible with other refinements of resolution, in particular ordering-based restrictions, in the sense that their combination is not complete. As state-of-the-art provers strongly rely on such restrictions to limit their search space, we cannot use the set-of-support strategy to integrate theories into them.

Another way to handle theories is deduction modulo [2]. In deduction modulo, the theory is presented by means of a congruence over propositions, the inference rules of existing proof systems being applied modulo this congruence. In practice, this congruence is often defined by a rewrite system that can rewrite not only terms into terms but also atomic propositions into general propositions. Corresponding proof search methods are then obtained by combining an existing method with narrowing[1]. Thus, there are proof-search procedures extending resolution, such as ENAR [2] or its more recent version called Polarized Resolution Modulo [3]; or extending tableaux methods [4]. Examples of theories that can be used in deduction modulo include arithmetic [5], Zermelo's set theory [6], simple type theory (a.k.a. higher-order logic) [7] or pure type systems [8, 9], and there exists a procedure to present any first-order classical theory as a rewrite system usable in deduction modulo [10].

Depending on the rewrite system presenting the theory, proof search methods based on deduction modulo are not always complete. It can be proved that their completeness is equivalent to the admissibility of the cut rule in the sequent calculus modulo the rewrite system [11]. What can be seen as a drawback is in fact their strength. Indeed, the completeness of these methods implies the consistency of the theory represented by the rewrite system. Therefore, as a consequence of Gödel's incompleteness theorem (provided the theory is at least as strong as arithmetic), completeness cannot be proved in that theory itself. In particular, this shows that polarized resolution modulo is not an instance of known refinements of resolution [12], whose completeness can be proved in simple type theory. To prove the completeness of polarized resolution modulo, we therefore proceed in two steps: we first prove completeness with regard to the cut-free fragment of the sequent calculus modulo for any rewrite system. Then, for some particular rewrite system, we prove that cut admissibility holds, that is, the cut-free fragment corresponds to the whole. Due to Gödel's theorem, this proof has to be done in a stronger theory than the one defined by the rewrite system. A bunch of techniques exists to prove cut admissibility in deduction modulo [13–15] (in particular, they can be applied to the theories cited before, for which proof search methods modulo are therefore complete) and a completion procedure was designed to transform a rewrite system so that cut admissibility holds [10].

Instead of implementing polarized resolution modulo from scratch, we would like to embed it into existing provers. Two points need to be overcome: First,

[1] Meta-variables may need to be instantiated before being rewritten, hence the use of narrowing and not merely rewriting.

existing provers are not based on general resolution, but on some refinement of it. We therefore need to check if narrowing is compatible with these refinements. In particular, we have to know if polarized resolution modulo with ordering-based restrictions (as in ordered resolution) is still complete. In this paper, we define Ordered Polarized Resolution Modulo, and we prove its completeness relatively to the cut-free fragment. Quiet surprisingly, no compatibility between the rewrite system and the ordering is requested to ensure this completeness. We are also concerned with simplification rules, and we propose a general framework, based on a proof ordering, to show that some simplification rules preserve the completeness. We apply it to Strict Subsumption Elimination and Demodulation. Second, we need to see how to proceed from an implementing point of view. It turns out that seeing polarized resolution modulo as a combination of the set-of-support strategy and literal selection makes it easy to incorporate into provers based on a variant of the given-clause algorithm, as is the case for most of them.

The next section will present the minimal knowledge needed on deduction modulo to make the paper self-contained. In Section 2 we define the Ordered Polarized Resolution Modulo and prove its completeness. Section 3 introduces the ordering-based framework for completeness-preserving simplification rules, and applies it to Strict Subsumption Elimination and Demodulation. In Section 4, we discuss how the given-clause algorithm can be used to embed polarized resolution modulo into a prover.

1 Deduction Modulo

We use standard definitions for terms, predicates, propositions (with connectors $\neg, \Rightarrow, \wedge, \vee$ and quantifiers $\forall, \exists$), sequents, substitutions, term rewrite rules and term rewriting, as can be found in [16, 17]. $\mathcal{V}$ is the set of variables, the replacement of a variable x by a term t in a term or a proposition A is denoted by $\{t/x\}A$, the application of a substitution σ in a term or a proposition A by σA. To ensure the existence of ground terms (terms without free variable), we assume the existence of at least one constant. A term t can be narrowed into s using substitution σ at position $\mathfrak{p}$ ($t \overset{\mathfrak{p},\sigma}{\rightsquigarrow} s$) if σt can be rewritten to s using substitution σ at position $\mathfrak{p}$. A literal is an atomic proposition or the negation of an atomic proposition. A proposition is in clausal normal form if it is the universal quantification of a disjunction of literals $\forall x_1, \ldots, x_n.\ L_1 \vee \ldots \vee L_p$ where $x_1, \ldots, x_n$ are the free variables of $L_1, \ldots, L_p$. A multiset of propositions is in clausal normal form if all its elements are. In the following, we will often omit to write the quantifications, and we will identify propositions in clausal normal form with clauses (i.e. set of literals) as if $\vee$ was associative, commutative and idempotent. Justifications for this will be given later. $\Box$ represents the empty clause. The polarity of a position in a proposition can be defined as follow: the root is positive, and the polarity switches when going under a $\neg$ or on the left of a $\Rightarrow$. x, y, z, u ranges over variables, s, t over terms, A, B over propositions, C, D, E, F over clauses or propositions in clausal normal form, L, K over literals, P, Q over atomic propositions and Γ, Δ over multisets of propositions.

$$\frac{}{\Gamma, A \vdash B, \Delta}\ A \xrightarrow[\mathcal{R}]{*} {}^{-}C\ {}^{+}\xleftarrow[\mathcal{R}]{*} B \qquad \frac{\Gamma, A \vdash \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash \Delta}\ A\ {}^{-}\xleftarrow[\mathcal{R}]{*} C \xrightarrow[\mathcal{R}]{*} {}^{+}B$$

$$\vee\vdash \frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, C \vdash \Delta}\ C \xrightarrow[\mathcal{R}]{*} {}^{-}A \vee B \qquad \vdash\exists \frac{\Gamma \vdash \{t/x\}A, \Delta}{\Gamma \vdash B, \Delta}\ B \xrightarrow[\mathcal{R}]{*} {}^{+}\exists x.\ A$$

Fig. 1. Some inference rules of the Polarized Sequent Calculus Modulo $\mathcal{R}$

In deduction modulo, term rewriting and narrowing is extended to propositions by congruence. In addition, there are also proposition rewrite rules whose left hand side is an atomic proposition and whose right hand side can be any proposition. Such rules can also be applied to non-atomic propositions by congruence. It can be useful to distinguish whether a proposition rewrite rule can be applied at a positive or a negative position. To do so, proposition rewrite rules are tagged with a polarity; they are then called polarized rewrite rules. A proposition A is rewritten positively into a proposition B ($A \longrightarrow {}^{+}B$) if it is rewritten by a positive rule at a positive position or by a negative rule at a negative position. It is rewritten negatively ($A \longrightarrow {}^{-}B$) if it is rewritten by a positive rule at a negative position or by a negative rule at a positive position. Term rewrite rules are considered as both positive and negative. $\xrightarrow{*}{}^{\pm}$ is the reflexive transitive closure of $\longrightarrow {}^{\pm}$.

In deduction modulo [2], the inference rules of an existing system such as the sequent calculus are applied modulo the congruence associated with the rewrite system (term rewrite rules and proposition rewrite rules). This leads for instance to the sequent calculus modulo. In polarized deduction modulo [18], polarities of rewrite rules are also taken into account. Some inference rules of the polarized sequent calculus modulo are presented in Figure 1. We write $\Gamma \vdash_{\mathcal{R}} \Delta$ if the sequent $\Gamma \vdash \Delta$ can be proved in the polarized sequent calculus modulo $\mathcal{R}$, and $\Gamma \vdash^{cf}_{\mathcal{R}} \Delta$ if it can be proved without the cut rule ($\frac{}{\vdash}$).

In the original version of (polarized) deduction modulo, term rewrite rules are taken into account as an equational theory $\mathcal{E}$. In the extension of the resolution method based on deduction modulo, this is performed by using unification modulo $\mathcal{E}$ instead of syntactical unification, following the equational resolution [19] where unification constraints are used instead of substitutions. In addition to this Resolution rule, an Extended Narrowing rule permits to narrow propositions using the proposition rewrite rules. The Polarized Resolution Modulo is presented in Figure 2. Note that the polarized rewrite system $\mathcal{R}$ is assumed to be clausal, that is, the right hand side of negative rewrite rules is a proposition in clausal normal form, and the one of positive rules is the negation of a clausal normal form. This ensures that narrowed propositions stay in clausal normal form. To see some examples, in particular how higher-order logic is used in Polarized Resolution Modulo, see [3]. However, note that we do not rely on any result of [3] in this paper.

As far as the author knows, no efficient first-order theorem prover uses such constraints. It is indeed not trivial to implement them while avoiding clauses with unsatisfiable constraints. Instead, term indexing is used to reduce the number

$$\textsf{Resolution}\ \frac{P_1 \vee \ldots \vee P_n \vee C \cdot [\mathfrak{C}_1] \qquad \neg Q_1 \vee \ldots \vee \neg Q_p \vee D \cdot [\mathfrak{C}_2]}{C \vee D \cdot [\mathfrak{C}_1 \cup \mathfrak{C}_2 \cup \{P_1 =^?_{\mathcal{E}} \cdots =^?_{\mathcal{E}} P_n =^?_{\mathcal{E}} Q_1 =^?_{\mathcal{E}} \cdots =^?_{\mathcal{E}} Q_p\}]}$$

$$\textsf{Ext. Narr.}\ \frac{P \vee C \cdot [\mathfrak{C}]}{D \vee C \cdot [\mathfrak{C} \cup \{P =^?_{\mathcal{E}} Q\}]}\ Q \to D \text{ is a negative rule in } \mathcal{R}$$

$$\textsf{Ext. Narr.}\ \frac{\neg P \vee C \cdot [\mathfrak{C}]}{D \vee C \cdot [\mathfrak{C} \cup \{P =^?_{\mathcal{E}} Q\}]}\ Q \to \neg D \text{ is a positive rule in } \mathcal{R}$$

Fig. 2. Inference rules of the Polarized Resolution Modulo $\mathcal{R}, \mathcal{E}$ ($\mathrm{PRM}_{\mathcal{R},\mathcal{E}}$)

of clauses that are candidates for resolution. To get closer from the implementation, the idea would therefore be to adapt term indexing techniques to equational unification. However, as far as the author knows, no generic term indexing modulo exists, only term indexing for some particular theories such as AC or HOL. Instead, for want of a better solution, assuming that the equational theory is presented as a set of term rewrite rules, we will apply Extended Narrowing using these rules also:

$$\textsf{Ext. Narr.}\ \frac{L \vee C}{\sigma(L' \vee C)}\ L \overset{\mathfrak{p};\sigma}{\underset{\mathcal{E}}{\rightsquigarrow}} L',\ L_{|\mathfrak{p}} \notin \mathcal{V}$$

2 Refining Polarized Resolution Modulo

Literal selections in clauses permit to restrict the application of Resolution. In this section, we show that using an ordering-based literal selection preserves the completeness of $\mathrm{PRM}_{\mathcal{R}}$. We use an ordering $\succ$ on literals which is well-founded and stable by substitution, and we assume that $\succ$ can be extended to an ordering $\succ_g$ that is total on ground literals. Note that it is more general than starting from an ordering on atoms and extending it to literals, that is, following the terminology of [20], we use a L-ordering and not a A-ordering (furthermore, we do not require that $P \not\succ \neg P$ for all atoms P). Rules of the Ordered Polarized Resolution Modulo ($\mathrm{OPRM}^{\succ}_{\mathcal{R}}$) are presented in Figure 3. To stay nearer from the existing implementations of resolution-based proved, we do not use constraints, Resolution only uses one literal per clause and there is therefore a Factoring rule. We write $\Gamma \mapsto\!\!\!\!\!\rightarrow^{\succ}_{\mathcal{R}} C$ when a clause C can be derived from the set of clauses Γ using finitely many inference rules of $\mathrm{OPRM}^{\succ}_{\mathcal{R}}$.

We want to prove that any *cut-free* proof of the polarized sequent calculus modulo $\mathcal{R}$ can be transformed into a derivation of the empty clause in $\mathrm{OPRM}^{\succ}_{\mathcal{R}}$. [2, Lemma 4.1] shows that transforming a formula into its clausal normal form does not change its refutability. The results from [11, Section 5] shows that the order of the quantifiers and of the disjunctions is not relevant w.r.t. refutability, so that we can consider propositions in clausal normal form as clauses. Therefore, we can restrict ourselves to proof of sequents $\Gamma \vdash$ where Γ is in clausal normal form. The theorem we want to prove is then the following:

$$\textsf{Resolution}\ \frac{P \vee C \qquad \neg Q \vee D}{\sigma(C \vee D)}\ {}^{a,b,c} \qquad \textsf{Factoring}\ \frac{L \vee K \vee C}{\sigma(L \vee C)}\ {}^{d}$$

$$\textsf{Ext. Narr.}\ \frac{P \vee C}{\sigma(D \vee C)}\ {}^{a,b},\ Q \to^{-} D \qquad \textsf{Ext. Narr.}\ \frac{\neg Q \vee D}{\sigma(C \vee D)}\ {}^{a,c},\ P \to^{+} \neg C$$

$$\textsf{Ext. Narr.}\ \frac{L \vee C}{\sigma(L' \vee C)}\ {}^{e},\ L \overset{\mathfrak{p},\sigma}{\rightsquigarrow} L' \text{ by a term rewrite rule, } L_{|\mathfrak{p}} \notin \mathcal{V}$$

[a] $\sigma = mgu(P, Q)$
[b] P maximal in $P \vee C$
[c] $\neg Q$ maximal in $\neg Q \vee D$
[d] L and K maximal in $L \vee K \vee C$, $\sigma = mgu(L, K)$
[e] L maximal in $L \vee C$

Fig. 3. Inference rules of the $\text{OPRM}^{\succ}_{\mathcal{R}}$

Theorem 1. *Given a set of clauses Γ, if $\Gamma \vdash^{cf}_{\mathcal{R}}$ then $\Gamma \mapsto\!\!\!\!\twoheadrightarrow^{\succ}_{\mathcal{R}} \square$.*

To get a less direct but more elegant proof, we use a couple of intermediary calculi. First, as in [2, 3], we need an intermediary resolution calculus where the instantiations are separated from the resolution and extended narrowing rules. This calculus, which is essentially PEIR [3] but with ordering restrictions, is called $\text{OPEIR}^{\succ}_{\mathcal{R}}$ for Ordered Polarized Extended Identical Resolution and is presented in Figure 4. We write $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} C$ when a clause C can be derived from the set of clauses Γ using finitely many inference rules of $\text{OPEIR}^{\succ}_{\mathcal{R}}$.

$$\textsf{Identical Resolution}\ \frac{P \vee C \qquad \neg P \vee D}{C \vee D}\ {}^{a} \qquad \textsf{Instantiation}\ \frac{C}{\{t/x\}C}$$

$$\textsf{Reduction}\ \frac{P \vee C}{D \vee C}\ {}^{a},\ P \longrightarrow^{-} D \qquad \textsf{Reduction}\ \frac{\neg P \vee D}{C \vee D}\ {}^{a},\ P \longrightarrow^{+} \neg C$$

$$\textsf{Reduction}\ \frac{L \vee C}{L' \vee C}\ {}^{b},\ L \longrightarrow L' \text{ by a term rewrite rule}$$

[a] P maximal in $P \vee C$ (resp. $\neg P$ in $\neg P \vee D$)
[b] L maximal in $L \vee C$

Fig. 4. Inference rules of the $\text{OPEIR}^{\succ}_{\mathcal{R}}$

We also need a sequent calculus modulo which is more adapted to our purpose. Following the ideas of [10], we introduce the one-sided polarized unfolding sequent calculus (short $1\text{PUSC}_{\mathcal{R}}$) where all formulæ are put in the left-hand side of the sequents, instantiations are ground, rewrite steps are explicit, and rewrit-

$$\widehat{\vdash}\ \frac{}{\Gamma, P, \neg P \vdash} \qquad \because\!\vdash\ \frac{\Gamma, C, C \vdash}{\Gamma, C \vdash}$$

$$\vee\!\vdash\ \frac{\Gamma, C \vdash \qquad \Gamma, D \vdash}{\Gamma, C \vee D \vdash} \qquad \forall\!\vdash\ \frac{\Gamma, \{t/x\}C \vdash}{\Gamma, \forall x.\ C \vdash}\ t \text{ ground}$$

$$\uparrow^{-}\vdash\ \frac{\Gamma, C \vdash}{\Gamma, P \vdash}\ P \longrightarrow^{-} C \qquad \uparrow^{+}\vdash\ \frac{\Gamma, C \vdash}{\Gamma, \neg P \vdash}\ P \longrightarrow^{+} \neg C$$

Fig. 5. Inference rules of the $1PUSC_{\mathcal{R}}$

ing and axioms can be applied to literals only. Its inference rules are presented in Figure 5. Note that there are no cut rule, so that $1PUSC_{\mathcal{R}}$ is restricted to the cut-free fragment of deduction modulo $\mathcal{R}$. We write $\Gamma \Vdash_{\mathcal{R}}$ when a sequent $\Gamma \vdash$ can be proved in $1PUSC_{\mathcal{R}}$.

To prove Theorem 1, we proceed as follow: a cut-free proof in the Polarized Sequent Calculus Modulo $\mathcal{R}$ is transformed into a proof in $1PUSC_{\mathcal{R}}$, which is transformed into an $OPEIR^{\succ}_{\mathcal{R}}$ derivation, which is transformed into an $OPRM^{\succ}_{\mathcal{R}}$ derivation.

Proposition 2. *For all set of clauses Γ, if $\Gamma \vdash^{cf}_{\mathcal{R}}$, then $\Gamma \Vdash_{\mathcal{R}}$.*

Proof. We need to prove that weakening is admissible, that we can make the rewriting explicits and that we can restrict $\widehat{\vdash}$, $\uparrow^{-}\vdash$ and $\uparrow^{+}\vdash$ to literals. The proof is the same as for [10, Proposition 7], except that we are here in a one-sided sequent calculus, which is not problematic since all negations are put down on the literal level.

We also need to prove that all instantiations can be ground. By induction on the proof structure. Recall that it is assumed that there exists some constant c, so that ground terms exist. If a $\forall\vdash$ in the Polarized Sequent Calculus Modulo instantiate a variable by a non-ground term t, then either the variables of this term are not instantiate in the proof above, in which case replacing them with c keeps the validity of the proof, or they are instantiated by some $\forall\vdash$. By induction hypothesis, they are instantiated by a ground term. One variable x may be instantiated by different ground terms $s^x_1, \ldots, s^x_n$ in the proof above, so we have to apply $\because\vdash$ before applying $\forall\vdash$ using each of the t where the variables x are instantiated by one of the s^x_i. □

Lemma 3. *Starting from a set of clauses Γ (so without unbound variables), if we have a proof of $\Gamma \vdash$ in $1PUSC_{\mathcal{R}}$, then the sequents in this proof do not contain unbound variables.*

Proof. By simple induction on the proof structure, using the fact that we only instantiate ground terms.

Lemma 4. *For all set of clauses Γ, for all ground clauses $C_1, \ldots, C_n$ and D such that the literals of D are smaller or equal to the maximal literals of C_i for $\succ_g$, if $\Gamma, C_1, \ldots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \Box$ and $\Gamma, D \hookrightarrow^{\succ}_{\mathcal{R}} \Box$ then $\Gamma, C_1 \vee D, \ldots, C_n \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \Box$.*

Proof. By lexicographic induction on the multiset extension of $\succ_g$ applied on D and the number of derivation steps in $\Gamma, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$. We rely on the fact that $\succ_g$ is total on ground literals.

We try to reproduce the derivation $\Gamma, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$ but replacing the C_i by $C_i \vee D$. Let C be the first clause produced in that derivation. There are two cases:

- C is produced using other clauses than one of the C_i. We can therefore derive C in $\Gamma, C_1 \vee D, \dots, C_n \vee D$.
  The derivation length of $\Gamma, C, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$ is strictly smaller than the derivation length of $\Gamma, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$. Of course $\Gamma, C, D \hookrightarrow^{\succ}_{\mathcal{R}} \square$. By induction hypothesis, we therefore have $\Gamma, C, C_1 \vee D, \dots, C_n \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$.
  Hence, $\Gamma, C_1 \vee D, \dots, C_n \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$.
- At least one of the parents of C is one C_i. As the literals of D are smaller or equal to those of C_i for $\succ_g$, the maximal elements for $\succ$ of C_i are included in those of $C_i \vee D$. We can therefore derive $C \vee D$ in $\Gamma, C_1 \vee D, \dots, C_n \vee D$. There are two cases:
  - The literals of D are less or equal to the maximal literals of C for $\succ_g$. The derivation length of $\Gamma, C_1, \dots, C_n, C \hookrightarrow^{\succ}_{\mathcal{R}} \square$ is strictly smaller than of $\Gamma, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$, so that by induction hypothesis, we have $\Gamma, C_1 \vee D, \dots, C_n \vee D, C \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$. Hence $\Gamma, C_1 \vee D, \dots, C_n \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$.
  - At least one of the literals in D is strictly greater than one of the maximal literals of C. As $\succ_g$ is total on ground literals, the literals in C are strictly smaller than the maximal literals of D.
    The derivation length of $\Gamma, C, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$ is strictly smaller than the derivation length of $\Gamma, C_1, \dots, C_n \hookrightarrow^{\succ}_{\mathcal{R}} \square$. Of course $\Gamma, C, D \hookrightarrow^{\succ}_{\mathcal{R}} \square$. By induction hypothesis, we therefore have $\Gamma, C, C_1 \vee D, \dots, C_n \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$.
    We have $\Gamma, C_1 \vee D, \dots, C_n \vee D, D \hookrightarrow^{\succ}_{\mathcal{R}} \square$ and $\Gamma, C_1 \vee D, \dots, C_n \vee D, C \hookrightarrow^{\succ}_{\mathcal{R}} \square$, and C is strictly less than D for the multiset extension of $\succ$, so that by induction hypothesis we have $\Gamma, C_1 \vee D, \dots, C_n \vee D, D \vee C \hookrightarrow^{\succ}_{\mathcal{R}} \square$.
    Hence $\Gamma, C_1 \vee D, \dots, C_n \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$. $\square$

Note 5. The lemma does not use any compatibility between the polarized rewrite rules and the ordering $\succ$. Indeed, polarized rewrite rules may increase the maximal literals in the clauses, but this does not break the completeness.

Proposition 6. *A proof of $\Gamma \vdash$ in $1PUSC_{\mathcal{R}}$ can be transformed into a derivation $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} \square$.*

Proof. By induction on the structure of the proof.

If the last rule is $\overset{\frown}{\vdash} \dfrac{}{\Gamma, P, \neg P \vdash}$ then we apply Identical Resolution on P and $\neg P$ to derive the empty clause.

If the last rule is $\cdot\cdot\vdash \dfrac{\Gamma, C, C \vdash}{\Gamma, C \vdash}$ then by induction hypothesis we have a derivation $\Gamma, C, C \hookrightarrow^{\succ}_{\mathcal{R}} \square$, which is also a derivation $\Gamma, C \hookrightarrow^{\succ}_{\mathcal{R}} \square$.

If the last rule is $\vee\vdash \dfrac{\Gamma, C \vdash \qquad \Gamma, D \vdash}{\Gamma, C \vee D \vdash}$ then by induction hypothesis we have $\Gamma, C \hookrightarrow^{\succ}_{\mathcal{R}} \square$ and $\Gamma, D \hookrightarrow^{\succ}_{\mathcal{R}} \square$. By Lemma 3, C and D are ground. Without

loss of generality, we can assume that all the atoms in D are less or equal to the maximal atoms in C for $\succ_g$ (else, exchange C and D since $\succ_g$ is total on ground literals), so that we can apply Lemma 4 to obtain a derivation of $\Gamma, C \vee D \hookrightarrow^{\succ}_{\mathcal{R}} \square$.

If the last rule is $\forall\vdash \dfrac{\Gamma, \{t/x\}C \vdash}{\Gamma, \forall x.\ C \vdash}$ then by induction hypothesis we obtain a derivation $\Gamma, \{t/x\}C \hookrightarrow^{\succ}_{\mathcal{R}} \square$. Using Instantiation we can derive $\{t/x\}C$ from $\forall x.\ C$ (recall that we identify clauses and propositions in clausal normal form). We therefore have a derivation of $\Gamma, \forall x.\ C \hookrightarrow^{\succ}_{\mathcal{R}} \square$.

If the last rule is $\uparrow^- \vdash \dfrac{\Gamma, C \vdash}{\Gamma, P \vdash}\ P \xrightarrow{-} C$ then by induction hypothesis we have a derivation $\Gamma, C \hookrightarrow^{\succ}_{\mathcal{R}} \square$. Using Reduction, we can derive C from P and get a derivation $\Gamma, P \hookrightarrow^{\succ}_{\mathcal{R}} \square$. The case of $\uparrow^+ \vdash$ is dual. $\square$

We now want to transform an $\text{OPEIR}^{\succ}_{\mathcal{R}}$ derivation into an $\text{OPRM}^{\succ}_{\mathcal{R}}$ one. The principal difficulty is that we may have instantiated literals to much before applying Identical Resolution, so that we cannot translate it directly into a Resolution with the appropriate mgu. Note that the Instantiations can be regrouped into a more general derived rule $\text{Instantiation}\ \dfrac{\overline{C}}{\overline{\sigma C}}$ for a substitution σ. We can also cut substitutions to transform $\text{Instantiation}\ \dfrac{\overline{C}}{\overline{\sigma\theta C}}$ into $\begin{array}{l}\text{Instantiation}\ \dfrac{\overline{C}}{\overline{\theta C}} \\ \text{Instantiation}\ \dfrac{\overline{\theta C}}{\overline{\sigma\theta C}}\end{array}$.

Lemma 7. *If σL is maximal in $\sigma(L \vee C)$ then L is maximal in $L \vee C$.*

Proof. As $\succ$ is stable by substitution, suppose that L is not maximal in $L \vee C$, then it means that some literal K of C is greater than L, but this implies that σK is greater than σL in $\sigma(L \vee C)$. $\square$

Note 8. This lemma is no longer true if the selection of literals in a clause is not stable by substitution. Such a stability condition is also required in [21].

Proposition 9. *If $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} \square$ then $\Gamma \twoheadmapsto^{\succ}_{\mathcal{R}} \square$.*

Proof. We prove a stronger result: if $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} C$ then there exist a clause C' and a substitution θ such that $\Gamma \twoheadmapsto^{\succ}_{\mathcal{R}} C'$ and $C = \theta C'$.

By induction on the derivation $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} C$. If the last step is Instantiation, the result is immediate by induction hypothesis.

If the last step is $\text{Identical Resolution}\ \dfrac{P \vee C_1 \qquad \neg P \vee C_2}{C_1 \vee C_2}$ then by induction hypothesis there exists C'_1, $P^1_1, \ldots, P^n_1$, θ_1 and C'_2, $P^1_2, \ldots, P^m_2$, θ_2 such that $\theta_1 C'_1 = C_1$, $\theta_1 P^i_1 = P$ and $\Gamma \twoheadmapsto^{\succ}_{\mathcal{R}} P^1_1 \vee \cdots \vee P^n_1 \vee C'_1$, and $\theta_2 C'_2 = C_2$, $\theta_2 P^i_2 = P$ and $\Gamma \twoheadmapsto^{\succ}_{\mathcal{R}} \neg P^1_2 \vee \cdots \vee \neg P^m_2 \vee C'_2$. All P^i_1 are unifiable, since they all can be instantiated to P. Let $\sigma^1_1 = mgu(P^1_1, P^2_1)$. As P is maximal in $P \vee C_1$, by Lemma 7, P^1_1 and P^2_1 are maximal in $P^1_1 \vee \cdots \vee P^n_1 \vee C'_1$. We can therefore apply Factoring to get $\sigma^1_1(P^1_1 \vee P^3_1 \vee \cdots \vee P^n_1 \vee C'_1)$. Again, by Lemma 7, $\sigma^1_1 P^1_1$ is maximal in $\sigma^1_1(P^1_1 \vee P^3_1 \vee \cdots \vee P^n_1 \vee C'_1)$. By repeating this process, we therefore obtain $\sigma_1(P^1_1 \vee C'_1)$ with $\sigma_1 = mgu(P^1_1, \ldots, P^n_1)$. All the same, we can derive $\sigma_2(\neg P^1_2 \vee C'_2)$ in $\text{OPRM}^{\succ}_{\mathcal{R}}$ with $\sigma_2 = mgu(P^1_2, \ldots, P^n_2)$.

$\sigma_1 P_1^1$ and $\sigma_2 P_2^1$ are unifiable since they can be instanciated to P. Let $\sigma = mgu(\sigma_1 P_1^1, \sigma_2 P_2^1)$. $\sigma_1 P_1^1$ (resp. $\sigma_2 \neg P_2^1$) is maximal in $\sigma_1(P_1^1 \vee C_1')$ (resp, in $\sigma_2(\neg P_2^1 \vee C_2')$) due to Lemma 7. We can therefore apply Resolution to obtain a derivation of $\sigma(\sigma_1 C_1 \vee \sigma_2 C_2)$ in $\mathrm{OPRM}_{\mathcal{R}}^{\succ}$. By definition of the mgu, there exists some θ_1' and θ_2' such that $\theta_1 = \theta_1'\sigma\sigma_1$ and $\theta_2 = \theta_2'\sigma\sigma_2$. Considering as usual that C and D contains distinct variables, we therefore have $C \vee D = \theta_1\theta_2(\sigma(\sigma_1 C_1 \vee \sigma_2 C_2))$.

The proof is similar if the last step is a Reduction step, using Factoring and Ext. Narr. We just need to take care when the subterm that is narrowed is a variable, in which case we do not need narrowing and we can use instantiation instead. □

Proof (of Theorem 1). By successively using Propositions 2, 6 and 9. □

Note 10. In [21], a syntactic proof of the completeness of several refinement of resolution, including ordered resolution, is given. This is done by seeing resolution derivations as proofs using only cuts, and by permuting cuts. Due to the incompleteness of Polarized Resolution Modulo when cuts cannot be eliminated, it is not clear whether this method could be extended to deal with Extended Narrowing. Furthermore, we prefer to show that a cut-free sequent calculus proof can be transformed directly into a derivation satisfying the ordering restrictions.

3 Clause simplifications

Clause simplifications reduce the search space by eliminating redundancies or by putting clauses in some normal form. Not all simplifications preserving the completeness of ordered resolution can be used in $\mathrm{OPRM}_{\mathcal{R}}^{\succ}$. For instance, the elimination of tautologies, i.e. clauses of the form $C \vee P \vee \neg P$, makes $\mathrm{OPRM}_{\mathcal{R}}^{\succ}$ no longer complete.

Example 11. Consider the rewrite system $\mathcal{R} : P \to^{+} \neg Q, P \to^{-} \neg Q$, and the ordering $\neg Q \succ Q \succ \neg P \succ P$. It can be proved that cut admissibility holds in the sequent calculus modulo $\mathcal{R}$. Hence $\mathrm{OPRM}_{\mathcal{R}}^{\succ}$ is complete. In particular, the empty clause can be derived from the set of clauses $\neg Q \vee P, Q \vee \neg P$. However, the only clause that can be generated from these is $P \vee \neg P$, which is then narrowed into $P \vee Q$, but which would be eliminated as a tautology. $\mathrm{OPRM}_{\mathcal{R}}^{\succ}$ with tautology deletion is therefore not complete.

We give here a general framework, and we show that it can be applied to usual simplifications such as subsumption elimination or demodulation.

3.1 $>$-valid simplification rules

A simplification rule is a schema of the form $\dfrac{C_1 \ \cdots \ C_n}{D_1 \ \cdots \ D_m}$ that must be interpreted by: If there are clauses of the form $C_1, \ldots, C_n$, they are replaced by the corresponding clauses $D_1, \ldots, D_m$. In other terms, a set of clauses $\Gamma, C_1, \ldots, C_n$

can be transformed don't-care non-deterministically into $\Gamma, D_1, \ldots, D_m$ in a derivation. To show that $\text{OPEIR}^{\succ}_{\mathcal{R}}$ remains complete when adding some set of simplification rules, we will rely on some ordering on $1\text{PUSC}_{\mathcal{R}}$ proofs:

Definition 12. *An ordering $>$ over $1PUSC_{\mathcal{R}}$ proofs is said completeness-preserving*

- *if it is compatible with subproofs, i.e. if p is a subproof of q then $q > p$;*
- *and it is well-founded.*

A simplification rule $\dfrac{C_1 \ \cdots \ C_n}{D_1 \ \cdots \ D_m}$ *is said valid according to $>$ if for all its instances and for all set of clauses Γ,*

1. *if $\Gamma, D_1, \ldots, D_m \vdash_{\mathcal{R}}$ then $\Gamma, C_1, \ldots, C_n \vdash_{\mathcal{R}}$;*
2. *if $\Gamma, C_1, \ldots, C_n \vdash_{\mathcal{R}}$ then $\Gamma, D_1, \ldots, D_m \vdash_{\mathcal{R}}$ with a strictly smaller proof with respect to $>$;*
3. *the rule is stable by substitution, that is, for all substitution θ,* $\dfrac{\theta C_1 \ \cdots \ \theta C_n}{\theta D_1 \ \cdots \ \theta D_m}$ *is also an instance.*

Condition 1 is needed to prove the soundness of the calculus with the simplification rules. Condition 2 implies its completeness. Condition 3 permits to extend completeness from $\text{OPEIR}^{\succ}_{\mathcal{R}}$ to $\text{OPRM}^{\succ}_{\mathcal{R}}$.

Proposition 13. *Given a completeness-preserving ordering $>$ and a set of simplification rules valid according to that ordering, a proof of $\Gamma \vdash$ in $1PUSC_{\mathcal{R}}$ can be transformed into a derivation $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} \square$ using the simplification rules.*

Proof. We prove this by induction on $>$. If Γ can be simplified into Γ', then using Condition 2, we can find a smaller proof of $\Gamma' \vdash$ w.r.t. $>$, on which we can apply the induction hypothesis. If Γ cannot be simplified, we use the same arguments than in the proof of Proposition 6, relying on the fact that $>$ is compatible with subproofs. □

Proposition 14. *Given a completeness-preserving ordering $>$ and a set of simplification rules valid according to that ordering, a derivation $\Gamma \hookrightarrow^{\succ}_{\mathcal{R}} \square$ using the simplification rules can be transformed into a derivation $\Gamma \twoheadrightarrow^{\succ}_{\mathcal{R}} \square$ using the simplification rules.*

Proof. We use the same proof as for Proposition 9, with this supplementary argument: if a simplification rule can be applied to the clauses derived in $\text{OPRM}^{\succ}_{\mathcal{R}}$, then stability by substitution (Condition 3) tells us that the simplification rule can be applied on the corresponding instances in $\text{OPEIR}^{\succ}_{\mathcal{R}}$. □

3.2 Application

In this section, we assume that the term rewrite system that we are working modulo is terminating and confluent, and that polarized rewrite rules and term rewrite rules commute: if $P \longrightarrow^{\pm} (\neg)C$ in one step and $P \xrightarrow{*} Q$ with the term

rewrite system, then there exists D such that $Q \longrightarrow^{\pm} (\neg)D$ in one step and $C \xrightarrow{*} D$ with the term rewrite system. The usual rewrite systems used in deduction modulo, such as the encoding of higher-order logic, have this property.

We want to prove that the following usual simplification rules are complete:

- Strict Subsumption Elimination: $\dfrac{C \qquad (\sigma C) \vee D}{C}$, D not empty;
- Demodulation: $\dfrac{C}{D}$ if $C \longrightarrow D$ by the term rewrite system.

Repetitively applying Demodulation permits to obtain the normal form w.r.t. the term rewrite system.

We use the following ordering over $1\mathrm{PUSC}_{\mathcal{R}}$ proofs. The *skeleton* of a proof is the tree corresponding to the proof where nodes are couples of the inference rule and the principal formula. We define the following ordering $\rhd$ over inference rules: $\vee\vdash \rhd \forall\vdash \rhd \cdot,\cdot\vdash$ and $\subsetneq \rhd \uparrow^{\pm}\vdash \rhd \cdot,\cdot\vdash$, and we order formulæ with the term rewrite system (which is assumed to be terminating). We define a precedence (also noted $\rhd$) as the lexical combination of this two orderings. This precedence is therefore well-founded. We define as $>$ the lexicographic combination of the RPO based on this precedence applied on the skeleton of the proof and of the subset relation applied to the conclusion of the proof.

Lemma 15. *$>$ is a completeness-preserving ordering.*

Proof. As the precedence is well-founded, so is the RPO [22], therefore $>$ is well-founded. Furthermore, since a RPO is a simplification ordering, subproofs are indeed smaller according to $>$. □

Proposition 16. *Strict Subsumption Elimination and Clause Normalization are compatible with $>$.*

Proof. It is not hard to check that Condition 3 holds for these two rules.

Strict Subsumption Elimination: Condition 1 is a consequence of weakening as for Tautology Deletion. For Condition 2, we need to be more precise on the free variables of C and $(\sigma C) \vee D$. Let $x_1, \ldots, x_n$ be the free variables of C that are in the support of σ, and $z_1, \ldots, z_l$ the others free variables of C. Let $y_1, \ldots, y_m$ be the variables in the image of σ. Let $u_1, \ldots, u_k$ be the free variables of D not in $z_1, \ldots, z_l, y_1, \ldots, y_m$. We therefore want to prove that a proof p of $\Gamma, \forall x_1, \ldots, x_n, z_1, \ldots, z_l.\ C, \forall y_1, \ldots, y_m, z_1, \ldots, z_l, u_1, \ldots, u_k.\ \sigma(C \vee D) \vdash$ can be transformed into a smaller proof of $\Gamma, \forall x_1, \ldots, x_n, z_1, \ldots, z_l.\ C \vdash$. The idea is to follow the skeleton of p, except that we do not apply the instantiations of $y_1, \ldots, y_m$ and $u_1, \ldots, u_k$, and we replace the applications of $\vee\vdash$ such as $\vee\vdash \dfrac{\overset{\pi_1}{\Gamma', \theta\sigma C \vdash} \qquad \overset{\pi_2}{\Gamma', \theta D \vdash}}{\Gamma', \theta\sigma C \vee \theta D \vdash}$ by $\forall\vdash \dfrac{\overset{\pi_1}{\Gamma', \theta\sigma C \vdash}}{\Gamma', \theta C \vdash}$ where we instantiate the x_i by $\theta\sigma x_i$. We obtain this way a proof of $\Gamma, \forall x_1, \ldots, x_n, z_1, \ldots, z_l.\ C, \forall x_1, \ldots, x_n, z_1, \ldots, z_l.\ C \vdash$ on which we apply $\cdot,\cdot\vdash$. As $\vee\vdash \rhd \forall\vdash \rhd \cdot,\cdot\vdash$ and $\uparrow^{\pm}\vdash \rhd \cdot,\cdot\vdash$, we can verify that the skeleton of the proof that we obtain is strictly smaller than the one of the original proof (or $(\sigma C) \vee D$ is not used, and there are less propositions in the conclusion).

Clause Normalization: For Condition 1, we need to add the term rewriting into the proof. Since $\uparrow^{\pm}\vdash$ can only be applied to literals, we have to postpone the rewriting to the places where we use literals, which is not problematic. For Condition 2, we prove the stronger result that if $C \xrightarrow{*} D$ with the term rewrite system and $\Gamma, C \vdash_{\mathcal{R}}$, then we can find a smaller proof of $\Gamma, D \vdash$. We proceed by induction on $\longrightarrow$ applied to C (recall that the term rewrite system is supposed terminating), and on the proof structure. The most interesting cases are for $\frown\vdash$ and $\uparrow^{\pm}\vdash$. For $\frown\vdash$, suppose that we have $\frown\vdash \dfrac{}{\Gamma, \neg C, C \vdash}$. Then to get a proof of $\Gamma, \neg C, D \vdash$, we first need to apply (possibly several times) $\uparrow^{+}\vdash$ on $\neg C$ to obtain $\neg D$, which is possible since term rewrite rules have no polarity. After that we can apply $\frown\vdash$. As $(\frown\vdash, C) \rhd (\frown\vdash, D)$ and $\frown\vdash \rhd \uparrow^{+}\vdash$, we can check that the skeleton of the resulting proof is indeed smaller. For $\uparrow^{\pm}\vdash$, suppose that we have $\uparrow^{-}\vdash \dfrac{\overset{\pi}{\Gamma, E \vdash}}{\Gamma, C \vdash}$ with $C \xrightarrow{-} D$. If it is a polarized rewrite rule that is used, we use the fact that polarized rules and term rewrite rules commute to obtain some F such that $D \longrightarrow {}^{-}F \xleftarrow{*} E$. By induction hypothesis, we can obtain a proof π' of $\Gamma, F \vdash$ smaller than π. Then, $\uparrow^{-}\vdash \dfrac{\overset{\pi'}{\Gamma, F \vdash}}{\Gamma, D \vdash}$ is smaller than the first proof. If it is a term rewrite rule that is used, we can proceed similarly using the confluence of the term rewrite system instead of the commutation. D may need several $\uparrow^{-}\vdash$ steps to be rewritten into F, but these steps will be smaller for $\rhd$ than the step for $C \longrightarrow E$, therefore the resulting proof will be smaller. □

Corollary 17. *$\mathrm{OPRM}^{\succ}_{\mathcal{R}}$ with Strict Subsumption Elimination and Demodulation is complete, provided cut admissibility holds for $\mathcal{R}$.*

Note that we cannot hope to have completeness of full subsumption (that is, with D possibly empty) in $\mathrm{OPEIR}^{\succ}_{\mathcal{R}}$ since it would make Instantiation useless. It is not clear whether it is complete or not to eliminate full subsumptions in $\mathrm{OPRM}^{\succ}_{\mathcal{R}}$.

Note 18. In [23], an ordering is also used to give a syntactic proof of the completeness of ordered resolution with some simplification rules. However, this ordering has to be defined on propositions and is then extended to proofs (which are in that case resolution derivations), and the same ordering is used for the literal selection and for the validity of the simplification rules. In our framework, $\succ$ and $>$ can be completely independent.

4 Implementation Issues

We have seen that polarized resolution modulo is compatible with the ordering restrictions and some simplification rules present in the calculi on which automated provers are based. In this section, we look at how, in practice, polarized resolution modulo could be integrated in them.

The given-clause algorithm is used to organize which clauses must be used by inference rules in a automated prover. It is originally based on the set-of-support strategy for the resolution [1]. Depending on which clauses are simplified, there exists (at least) two variants of this algorithm, the Otter and the Discount loops, named after the prover in which they appeared. Most of today's automated provers are based on one of these variants. To keep it simple, we will only present the basic given-clause algorithm, without simplification rules.

The proof space is separate into two sets of clauses: the first one contains the set-of-support clauses, also called passive clauses, also called unprocessed clauses; the other one in the set of usable clauses, also called active clauses. At each step of the loop, a clause, called the given clause, is extracted from the set of passive clauses and put into the set of active clauses. All inferences between the given clause and one of the active clauses (comprising the given clause itself) are performed, the generated clauses being put into the set of passive clauses. At the beginning, the set of active clauses is therefore empty, and the clauses we want to refute or prove satisfiable are put into the set of passive clauses. Given a fair choice of the given clause, completeness of such an algorithm is not hard to prove. This algorithm has been proved quite successful because the set of active clauses can be organized in order to restrict the clauses where to apply the inference rules. In particular, active clauses are put into a term index, often based on discrimination trees, to make the retrieval of clauses containing literals potentially unifiable with the complement of some literal more efficient.

When no literal selection is used and we know that some subset of the input clauses is consistent, we can put directly this subset of clauses into the set of active clauses. Completeness is ensured by the completeness of the set-of-support strategy. However, set of support is no longer complete when using literal selection, even for selection based on some ordering. Actually, it can be proved that the completeness of set of support with selection requires a stronger property than the consistency of the theory, namely the admissibility of the cut rule in the sequent calculus modulo the theory. Indeed, Dowek [3] has shown that polarized rewrite rules can be seen as special clauses, that he called one-way clauses, in which one literal only is selected, and which cannot be resolved one with the other. More precisely, a positive rule $P \rightarrow^{+} \neg C$ corresponds to the clause $\underline{P} \vee C$, and a negative rule $P \rightarrow^{-} C$ to the clause $\underline{\neg P} \vee C$ (selected literals are underlined). Then, using Resolution with one of these one-way clauses correspond exactly to using Extended Narrowing with the associated polarized rewrite rule, and this way, one-way clauses are not resolved between themselves. To simulate Extended Narrowing into a prover, the idea is therefore to add the one-way clauses corresponding to the polarized rewrite rules directly into the set of active clauses, with their selected literal, and to put the input clauses as usual in the set of passive clauses.

We have applied these ideas to integrate polarized resolution modulo into the resolution prover included in `iprover` [24]. It would have been harder to integrate them into a prover based on superposition, because in these provers, selection is not symmetrical between positive and negative literals. We tested it using

the encoding in deduction modulo of the problems of higher-order logic of the TPTP [25]. As can be expected, performances compared to the provers dedicated to higher-order logic such as TPS (`http://gtps.math.cmu.edu/tps.html`) is quite poor, but they are promising (about a third of the problems solved by TPS can be solved by the modified `iprover`). We need to fine-tune the prover (for instance by choosing convenient orderings) to adapt it to HOL, and to look at other theories.

5 Conclusion and Discussion

We have shown how polarized deduction modulo can be embedded into an existing resolution-based prover and we have proved that ordered polarized resolution modulo with strict subsumption elimination and demodulation is complete. For this, we have defined an ordering-based criterion that ensures that simplification rules preserve completeness. Note that this criterion can also be used for standard resolution. These results are the first which lead to an actual and useful implementation of deduction modulo, and can be used to get automated theorem provers adapted to many theories, including arithmetic, Zermelo's set theory and higher-order logic. We are currently investigating whether using these refinements induces decision procedures for some classes of propositions, as it is the case for standard resolution [26].

Also, the treatment of equality in deduction modulo may be improved, because deduction modulo is originally based on proof systems without equality. Theoretically, this is not a problem, because the equality predicate can be encoded using a rewrite system representing Leibniz's axiom schema $x = y \Rightarrow A(x) \Rightarrow A(y)$. However, in practice, this encoding is not well suited, because the proposition A has to be guessed using narrowing steps. A solution would be to have a proof-search procedure modulo for first-order logic with equality, for instance by adding Extended Narrowing in the superposition calculus. It remains to be proved that the restrictions on the inference rules and the redundancy eliminations of superposition can be mixed with Extended Narrowing without breaking completeness. The usual proof of completeness of superposition relies on saturation up to redundancies w.r.t. $\succ$ [27, 28]. If we want to adapt this proof directly, we have to require that the one-way clauses corresponding to the rewrite rules are saturated for $\succ$. On the contrary of what is done here, this creates a dependency between the rewrite system and $\succ$. It can be proved that this assumption implies the cut admissibility [13], which explains why we would have full completeness in that case. However, we would like to drop this assumption for at least two reasons: First, some rewrite systems are not compatible with any well-founded, stable by substitution ordering, although cut admissibility holds for them. We therefore conjecture that superposition modulo is fully complete for those systems too. Second, even if a rewrite system is compatible with some ordering, the results of this paper show that another ordering can be used while remaining complete, thus offering new perspectives on combining orderings.

References

1. Wos, L., Robinson, G.A., Carson, D.F.: Efficiency and completeness of the set of support strategy in theorem proving. J. ACM **12** (1965) 536–541
2. Dowek, G., Hardin, T., Kirchner, C.: Theorem proving modulo. Journal of Automated Reasoning **31** (2003) 33–72
3. Dowek, G.: Polarized resolution modulo. Manuscript, available on the author's web page (2009)
4. Bonichon, R., Hermant, O.: A semantic completeness proof for TaMed. In Hermann, M., Voronkov, A., eds.: LPAR. Volume 4246 of LNCS., Springer (2006) 167–181
5. Dowek, G., Werner, B.: Arithmetic as a theory modulo. In Giesl, J., ed.: RTA. Volume 3467 of LNCS., Springer (2005) 423–437
6. Dowek, G., Miquel, A.: Cut elimination for Zermelo's set theory. Available on authors' web page (2006)
7. Dowek, G., Hardin, T., Kirchner, C.: HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. Mathematical Structures in Computer Science **11** (2001) 1–25
8. Cousineau, D., Dowek, G.: Embedding pure type systems in the lambda-pi-calculus modulo. In Ronchi Della Rocca, S., ed.: TLCA. Volume 4583 of LNCS., Springer (2007) 102–117
9. Burel, G.: A first-order representation of pure type systems using superdeduction. In Pfenning, F., ed.: LICS, IEEE Computer Society (2008) 253–263
10. Burel, G., Kirchner, C.: Regaining cut admissibility in deduction modulo using abstract completion. Inform. Comput. **208** (2010) 140–164
11. Hermant, O.: Resolution is cut-free. Journal of Automated Reasoning **44** (2009) 245–276
12. Burel, G., Dowek, G.: How can we prove that a proof search method is not an instance of another? In: LFMTP'09. ACM International Conference Proceeding Series, ACM (2009) 84–87
13. Hermant, O.: Méthodes Sémantiques en Déduction Modulo. PhD thesis, École Polytechnique (2005)
14. Dowek, G., Werner, B.: Proof normalization modulo. The Journal of Symbolic Logic **68** (2003) 1289–1316
15. Dowek, G.: Truth values algebras and proof normalization. In Altenkirch, T., McBride, C., eds.: TYPES. Volume 4502 of LNCS., Springer (2006) 110–124
16. Baader, F., Nipkow, T.: Term *Re*writing and *all That*. Cambridge University Press (1998)
17. Gallier, J.H.: Logic for Computer Science: Foundations of Automatic Theorem Proving. Volume 5 of Computer Science and Technology Series. Harper & Row, New York (1986) Revised On-Line Version (2003), `http://www.cis.upenn.edu/~jean/gbooks/logic.html`.
18. Dowek, G.: What is a theory? In Alt, H., Ferreira, A., eds.: STACS. Volume 2285 of LNCS., Springer (2002) 50–64
19. Plotkin, G.: Building in equational theories. In Meltzer, B., Michie, D., eds.: Machine Intelligence. Volume 7., Edinburgh, Scotland, Edinburgh University Press (1972) 73–90
20. de Nivelle, H.: A unification of ordering refinements of resolution in classical logic. In MacNish, C., Pearce, D., Pereira, L.M., eds.: JELIA. Volume 838 of LNCS., Springer (1994) 217–230

21. Goubault-Larrecq, J.: A note on the completeness of certain refinements of resolution. Research Report LSV-02-8, Laboratoire Spécification et Vérification, ENS Cachan, France (2002) 16 pages.
22. Dershowitz, N.: Orderings for term-rewriting systems. Theoretical Computer Science **17** (1982) 279–301
23. Bachmair, L.: Proof normalization for resolution and paramodulation. In Dershowitz, N., ed.: Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA). Volume 355 of LNCS., Springer (1989) 15–28
24. Korovin, K.: iProver – an instantiation-based theorem prover for first-order logic (system description). In Armando, A., Baumgartner, P., eds.: IJCAR. Volume 5195 of LNAI., Springer (2008) 292–298
25. Sutcliffe, G., Benzmüller, C., Brown, C.E., Theiss, F.: Progress in the development of automated theorem proving for higher-order logic. In Schmidt, R.A., ed.: CADE. Volume 5663 of LNCS., Springer (2009) 116–130
26. Joiner, Jr., W.H.: Resolution strategies as decision procedures. J. ACM **23** (1976) 398–417
27. Bachmair, L., Ganzinger, H.: Rewrite-based equational theorem proving with selection and simplification. J. Log. Comput. **4** (1994) 217–247
28. Ganzinger, H., Stuber, J.: Superposition with equivalence reasoning and delayed clause normal form transformation. Inf. Comput. **199** (2005) 3–23