Theorem Proving Modulo

Gilles Dowek, Thérèse Hardin, Claude Kirchner

Theorem Proving Modulo

Gilles Dowek, Thérèse Hardin, Claude Kirchner

No 3400

April 1998

THÈME 2

N 0249-6399

Theorem Proving Modulo

Gilles Dowek[*], Thérèse Hardin[†], Claude Kirchner[‡]

Thème 2 — Génie logiciel
et calcul symbolique
Projets Coq, Para, Protheo

Rapport de recherche n ° 3400 — April 1998 — 27 pages

Abstract: "Theorem proving modulo" is a way to remove computational arguments from proofs by reasoning modulo a congruence on propositions. Such a technique, issued from automated theorem proving, is of wider interest because it aims at separating deductions and computations. The first contribution of this paper is to provide a "sequent calculus modulo" that gives a clear distinction between the decidable (computation) and the undecidable (deduction).

The congruence on propositions is handled via rewrite rules and equational axioms. Usually rewriting applies only to terms. The second contribution of this paper is to allow rewriting atomic propositions into non atomic ones and to give a complete proof search method, called "Extended Narrowing and Resolution" (ENAR), modulo such congruences. The completeness of this method is proved using the sequent calculus modulo.

An important application is that this Extended Narrowing and Resolution method subsumes full higher-order resolution when applied to a first-order presentation of higher-order logic. This shows that such a presentation can yield also efficient proof-search methods.

Key-words: Automated theorem proving, rewriting, resolution narrowing, higher-order logic

(Résumé : tsvp)

[*] INRIA-Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France, Gilles.Dowek@inria.fr
[†] LIP6 and INRIA-Rocquencourt, Université Paris 6, 4, Place Jussieu, 75252 Paris Cedex 05, France, Therese.Hardin@lip6.fr
[‡] LORIA and INRIA, 615, rue du Jardin Botanique, B.P. 101, 54602 Villers-lès-Nancy Cedex, France, Claude.Kirchner@loria.fr

La Déduction modulo

Résumé : La “déduction modulo” est un moyen de supprimer les phases de calcul apparaissant dans les démonstrations en raisonnant modulo une congruence décrivant les calculs effectués sur les termes et les propositions. Cette technique, apparue en démonstration automatique, est d’une portée plus large car elle permet de séparer les phases de raisonnement et les phases de calcul. La première contribution de ce papier est la définition d’un “calcul des séquents modulo” qui sépare nettement le décidable (le calcul) de l’indécidable (le raisonnement).

La congruence sur les propositions est définie par un ensemble de règles de réécriture et d’axiomes équationnels. D’habitude, la réécriture s’applique uniquement aux termes. La seconde contribution de ce papier est d’étendre la réécriture aux propositions atomiques et de donner une méthode complète de démonstration automatique, appellée “Surréduction et résolution étendues” (ENAR), modulo de telles congruences. La complétude de cette méthode est établie en utilisant le calcul des séquents modulo.

Une application importante concerne la résolution d’ordre supérieur. Celle-ci peut être vue comme l’application de la méthode de surréduction et résolution étendues à une présentation au premier ordre de la logique d’ordre supérieur. La réduction au premier ordre peut donc également mener à des méthodes efficaces de démonstration automatique à l’ordre supérieur.

Mots-clé : Démonstration automatique, réécriture, résolution, surréduction, logique d’ordre supérieur

If let loose, usual automated theorem proving methods such as resolution or tableaux, can be very inefficient, spending time in trivial deductions. For instance with the associativity axiom:

$$\forall x\ \forall y\ \forall z\ ((x+y)+z = x+(y+z))$$

a prover may rearrange endlessly brackets. To improve efficiency, G. Plotkin [Plo72] proposes to drop the associativity axiom and to replace unification by equational unification modulo associativity (G. Plotkin says "building-in" the associativity axiom). This can be analyzed as first defining a quotient of the set of propositions by an appropriate congruence and then reformulating the proof search in the quotient.

This idea of working directly in a quotient of the set of propositions has appeared in automated theorem proving but is of wider interest and deserves to be studied for itself. Indeed defining a notion of proof in such a quotient yields a clear distinction between the notions of computation and deduction. The first contribution of this paper is to propose a sequent calculus modulo a congruence on propositions such that the propositions provable in this calculus are the same as the propositions provable in the usual sequent calculus extended by an appropriate theory depending on the congruence.

Usually the congruence on propositions is simply induced by a congruence on terms, but congruences defined directly on propositions are of prime interest, in particular when dealing with first-order presentations of higher-order logic.

Identifying atomic propositions with non atomic ones changes the logical structure of propositions. So, different deduction rules can be applied, according to the choice of the representative of an equivalence class. This leads to a true interaction between computation and deduction and this brings serious difficulties in automated theorem proving as well as in proof theory.

The second contribution of this paper is to introduce an *Extended Narrowing and Resolution* (ENAR) proof search method that extends equational resolution to a large class of congruences which allow to identify also atomic propositions with non atomic ones.

In particular, the ENAR method applied to a first-order presentation of higher-order logic subsumes higher-order resolution. Thus we show that expressing higher-order logic as a first-order theory and using a first-order proof search method is an efficient way to implement higher-order proof search, provided we use the right automated deduction tools for first-order logic.

After the end of this introduction, devoted to the presentation of the key ideas of the paper, Section 1 presents the *sequent calculus modulo* and its main properties. Section 2 defines the *Extended Narrowing and Resolution* (ENAR) proof search method and sections 3 and 4 are dedicated to the completeness proof of this method. We conclude in Section 5 by presenting some generalizations of the ENAR method.

On the use of congruences in automated theorem proving

As proposed by G. Plotkin, the explicit handling of equality in the deduction process can be avoided by integrating part of the reasoning modulo equality *in* the deduction process. This idea led a decade later to the so-called *theory resolution* of M.E. Stickel [Sti85].

As an example, in arithmetic, we may identify propositions equivalent modulo rearranging of brackets. This permits to drop the associativity axiom:

$$\forall x \; \forall y \; \forall z \; ((x+y)+z = x+(y+z))$$

as it belongs to the class of the trivial proposition:

$$\forall x \; \forall y \; \forall z \; (x+(y+z) = x+(y+z)).$$

In counterpart, unification, the basic operation of automated deduction, is no more the search of a substitution yielding identical terms but it has to find a substitution yielding congruent ones. That is, unification has to be replaced by *equational unification* [JK91].

In a similar way P. Andrews [And71] proposes to build-in conversion axioms of higher-order logic. Therefore unification is replaced by equational unification modulo $\beta\eta$-conversion, usually called higher-order unification [Hue72, Hue75].

The same concern of building-in part of the equality in equational reasoning itself [KB70] has also led to the study of equational reasoning modulo, whose main landmarks are the study of associative-commutative completion [PS81], the general study of coherence of an equational theory with respect to a rewrite system [JK86], its unified presentation in [Bac87] and its extension in [Vir95].

A drawback of this approach is that equational unification may be undecidable or quite complex. Fortunately deciding unifiability can be postponed and modularized by the use of constraints. Starting from the seminal work of G. Huet on higher-order resolution [Hue72], the notion of deduction with constraints spread, with constraint logic programming [JL87], and then constraint programming. The counterpart in theorem proving is deduction with constraint [KKR90] and complete constraint saturation processes [BGLS95]. This gives first a generalization of the concept of building-in specific theories, second it allows specific treatment of the constraints and third it generalizes to various constraint languages.

Finally, the various advances on rewriting techniques have demonstrated that orienting equality, i.e. determining which object is greater than the other, can lead to dramatical improvements. The integration of rewrite based techniques and ordering in first-order theorem proving began in the early eighties [HR86] (for a survey see [HKLR92]) leading to very powerful results and systems, which, together with the techniques based on constraints, allowed recently to solve problems considered by mathematicians as hard [McC97] (see also [Kol96]).

Identifying propositions

When considering rules rewriting propositions to propositions, equational resolution can easily be extended when both sides of the rewrite rules are atomic propositions. For instance, in arithmetic we can equate the propositions

$$S(x) = S(y) \text{ and } x = y$$

which allows to drop Peano's third axiom.

But we may also consider identifying atomic propositions with non atomic ones. A first attempt in this direction has been achieved by S.J. Lee and D. Plaisted [LP94] who have proposed a proof search method where defined predicate symbols may be unfolded.

Here we want to use more general rules transforming atomic propositions to arbitrary propositions. For instance, in the theory of integral rings, we want to have the equivalence between the propositions

$$x * y = 0 \text{ and } x = 0 \vee y = 0$$

or between

$$x * y = x * z \text{ and } y = z \vee x = 0$$

To operationalize the use of the congruence we consider a set $\mathcal{E}$ of equations and a set $\mathcal{R}$ of rewrite rules. Equations may relate terms to terms or atomic propositions to atomic ones.

Since propositions in first-order logic contain binders (quantifiers) such rewriting systems are in fact *Combinatory reduction systems* [KvOvR93]. As usual in first-order logic, we use a single category of names for free and bound variables.

Rewrite rules may rewrite atomic propositions to arbitrary ones. Rules rewriting terms to terms are added in the generalization presented in Section 5. Rules whose left members are not atomic such as the simplification of the proposition $A \wedge A$ to A are technically difficult to handle because putting a proposition in clause form may spread a redex. They are left outside the scope of this paper.

A typical example: HOL

Rules rewriting atomic propositions into non atomic ones are used for instance in (many-sorted) first-order formulations of higher-order logic (see, for example, [Dow97]).

The sorts of this theory are inductively defined by:
- ι and o are sorts,
- if T and U are sorts then $T \rightarrow U$ is a sort.

The language contains the constant symbols:
- $S_{T,U,V}$ of sort $(T \rightarrow U \rightarrow V) \rightarrow (T \rightarrow U) \rightarrow T \rightarrow V$,
- $K_{T,U}$ of sort $T \rightarrow U \rightarrow T$,
- $\dot{\neg}$ of sort $o \rightarrow o$,
- $\dot{\vee}$ of sort $o \rightarrow o \rightarrow o$,
- $\dot{\forall}_T$ of sort $(T \rightarrow o) \rightarrow o$,

the binary function symbol:
- $\alpha_{T,U}$ of rank $(T \rightarrow U, T)$ U

and the unary predicate symbol:

- ε of rank (o).

Notice that in this presentation there is a distinction between propositions as zero-ary relations (i.e. objects of type o) and true propositions of the first-order language. Objects of type o are built using the combinators $\dot{\neg}$, $\dot{\vee}$ and $\dot{\forall}_T$ while true propositions are built using the true connectors and quantifiers. If t is a term of type o, the corresponding proposition is written $\varepsilon(t)$ [Dow97].

The system $\mathcal{RE}$ is:

$$\mathcal{R} = \{\varepsilon(\alpha(\dot{\neg}, x)) \to \neg\varepsilon(x), \varepsilon(\alpha(\alpha(\dot{\vee}, x), y)) \to \varepsilon(x) \vee \varepsilon(y), \varepsilon(\alpha(\dot{\forall}_T, x)) \to \forall y\ \varepsilon(\alpha(x, y))\}$$

$$\mathcal{E} = \{\alpha(\alpha(\alpha(S, x), y), z) = \alpha(\alpha(x, z), (y, z)), \quad \alpha(\alpha(K, x), y) = x\}$$

The same kind of rules, in particular rules introducing quantifiers, can also occur in set theory, where we may want rules like:

$$\in (x, \{y, z\}) \to (x = y) \vee (x = z)$$

$$\in (x, \mathcal{P}(y)) \to \forall z\ (\in (z, x) \Rightarrow \in (z, y))$$

Extended narrowing

With such rules, the resolution method (or the tableaux method) is not complete anymore even if we replace unification by equational unification.

For instance, in the example of the integral rings, the proposition $\exists x\ (a * a = x \Rightarrow a = x)$ is provable since the propositions $a * a = 0$ and $a = 0 \vee a = 0$ are identified modulo the congruence:

$$\begin{array}{c}
\dfrac{a = 0 \vdash a = 0 \quad a = 0 \vdash a = 0}{a = 0 \vee a = 0 \vdash a = 0}\ \vee\text{-l} \\
\dfrac{\dfrac{a * a = 0 \vdash a = 0}{\vdash a * a = 0 \Rightarrow a = 0}\ \Rightarrow\text{-r}}{\vdash \exists x\ (a * a = x \Rightarrow a = x)}\ \exists\text{-r}
\end{array}$$

But, the clause form of its negation has only the clauses $a * a = x$ and $\neg a = x$ and thus the resolution rule cannot be applied.

As another example, in HOL, the proposition $\exists x\ \varepsilon(x)$ is provable since the propositions $\varepsilon(\alpha(\alpha(\dot{\vee}, y), \alpha(\dot{\neg}, y)))$ and $\varepsilon(y) \vee \neg\varepsilon(y)$ are identified modulo the congruence:

$$\begin{array}{c}
\dfrac{\dfrac{\varepsilon(y) \vdash \varepsilon(y)}{\vdash \varepsilon(y), \neg\varepsilon(y)}\ \neg\text{-r}}{\vdash \varepsilon(y) \vee \neg\varepsilon(y)}\ \vee\text{-r} \\
\dfrac{\vdash \varepsilon(\alpha(\alpha(\dot{\vee}, y), \alpha(\dot{\neg}, y)))}{\vdash \exists x\ \varepsilon(x)}\ \exists\text{-r}
\end{array}$$

But, the clause form of its negation has only the single clause $\neg\varepsilon(x)$ and thus the resolution rule cannot be applied.

This problem was already met in higher-order resolution [Hue72, Hue73], although in a different form because higher-order logic is not presented as a first-order theory there. It has led to the development of a new rule called **Splitting** which, together with the rule **Resolution**, form a complete system for higher-order logic. This **Splitting** rule allows to generate the different cases of proposition construction allowing to continue the search in a complete way. For example $\neg\varepsilon(x)$ will be splitted in $\neg\varepsilon(p_1), \ldots, \neg\varepsilon(p_n)$ according to the various possibilities to instantiate x in a minimal way.

In this paper, we show that such a rule can be formulated for a large class of first-order theories extended with a rewrite system rewriting atomic propositions to arbitrary ones. This rule happens to be quite similar to the narrowing rule used for equational unification [Hul80] and for combining logic and functional programming [Han94].

1 Sequent calculus modulo

In this section, we introduce a *sequent calculus modulo* a congruence which is operationally treated using a rewrite system and a set of equations. We establish the equivalence lemma stating that a proposition P is provable in this calculus if and only if it is provable in the standard sequent calculus using an appropriate set of axioms. However, although the two calculi have the same theorems, the proofs are different and the proofs modulo are shorter since computations steps are removed from them.

We consider a set of fixed arity function symbols $\mathcal{F}$, a set of fixed arity predicate symbols $\mathcal{P}$ and a set of variables $\mathcal{X}$ with no symbol in common. The set of terms built on $\mathcal{F}$ and $\mathcal{X}$ is denoted $\mathcal{T}(\mathcal{F}, \mathcal{X})$ or more generically `Term`. The set of atomic propositions built on the set of predicates $\mathcal{P}$ and the set of terms $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is denoted $\mathcal{AP}(\mathcal{P}, \mathcal{F}, \mathcal{X})$ and in a generic way `AtomProp`. The set of first-order propositions build on the atomic propositions in $\mathcal{AP}(\mathcal{P}, \mathcal{F}, \mathcal{X})$, with the usual connectors $\wedge, \vee, \Rightarrow, \neg, \bot$ and quantifiers $\forall$ and $\exists$ is denoted $\mathcal{P}(\mathcal{P}, \mathcal{F}, \mathcal{X})$ and in a generic way `Prop`.

We assume the reader familiar with the basic notions of term rewriting as described for example in [DJ90]. We are dealing in this work with congruences described by class rewrite systems, i.e. pairs composed of:
– $\mathcal{R}$: a rewrite system rewriting `AtomProp` to `Prop` and defining a rewrite relation denoted $\rightarrow_{\mathcal{R}}$,
– $\mathcal{E}$: a set of equational axioms equating `AtomProp` with `AtomProp` or `Term` with `Term` and defining a congruence denoted $=_{\mathcal{E}}$.

Remark 1.1 *It is possible, under some conditions, to consider additional rules in $\mathcal{R}$ rewriting also `Term` to `Term` (see section 5).*

Definition 1.1 *Given a class rewrite system $\mathcal{RE}$, the atomic proposition t $\mathcal{RE}$-*rewrites *to t', denoted $t \longrightarrow_{\mathcal{RE}} t'$, if $t =_{\mathcal{E}} u[\sigma(l)]_\omega$ and $t' =_{\mathcal{E}} u[\sigma(r)]_\omega$, for some rule $l \rightarrow r \in \mathcal{R}$,* *some atomic proposition u, some occurrence ω in u and some substitution σ. We denote $=_{\mathcal{RE}}$ the equality generated by $\mathcal{R}$ and $\mathcal{E}$. Following G. Peterson and M.E. Stickel, the atomic proposition t $\mathcal{R},\mathcal{E}$-rewrites to t', denoted $t \longrightarrow_{\mathcal{R},\mathcal{E}} t'$, if $t' = t[\sigma(r)]_{\omega}$, for some rule $l \to r \in \mathcal{R}$, some occurrence ω in t and some substitution σ such that $\sigma(l) =_{\mathcal{E}} t_{|\omega}$.*

For example, if $\mathcal{R} = \{a + b \to c\}$ and $\mathcal{E}$ contains the associativity and commutativity axioms, the term $b + (a + a)$ $\mathcal{RE}$-rewrites into $c + a$ but it is $\mathcal{R},\mathcal{E}$-irreducible.

In the rest of the paper, in order to let proof-checking be effective, we assume the following property of the relation $=_{\mathcal{RE}}$:

Assumption 1.1 *The relation $=_{\mathcal{RE}}$ to be decidable.*

Notice that neither $\to_{\mathcal{RE}}$ nor $\to_{\mathcal{R},\mathcal{E}}$ are supposed to be decidable. For instance in HOL, $=_{\mathcal{RE}}$ is decidable (see, for instance [Dow97]) but the decidablity of $\to_{\mathcal{RE}}$ seems to be linked to the decidability of higher-order matching which is, up to our knowledge, an open problem.

Definition 1.2 *We define in Figure 1 a sequent calculus, extending the usual one [GLT89, Gal86], by allowing working modulo $\mathcal{RE}$. In these rules we assume that the "," is associative and commutative, i.e. Γ and Δ are finite multisets of propositions.*

Remark 1.2 *Proof checking for the sequent calculus modulo is decidable: we can check for each rule that the conditions of application are satisfied because the relation $=_{\mathcal{RE}}$ is decidable and we provide the needed information in the quantifiers rules. When the congruence $=_{\mathcal{RE}}$ is simply the identity, this sequent calculus specializes to the standard one. In that case sequents are denoted using $\vdash$.*

Remark 1.3 *From the definition of the calculus, it is easy to see that if $P =_{\mathcal{RE}} Q$ then $\Gamma \vdash_{\mathcal{RE}} P, \Delta$ if and only if $\Gamma \vdash_{\mathcal{RE}} Q, \Delta$ and $\Gamma, P \vdash_{\mathcal{RE}} \Delta$ if and only if $\Gamma, Q \vdash_{\mathcal{RE}} \Delta$.*

Definition 1.3 *The theory $\mathcal{T}$ is said to be* compatible *with $\mathcal{RE}$ if, for any proposition P in $\mathcal{T}$ there is a proposition Q such that $P =_{\mathcal{RE}} Q$ and $\vdash Q$, i.e. P is equivalent to a proposition Q which is a tautology in the standard sequent calculus.*

Conversely, the class rewrite system $\mathcal{RE}$ is said to be compatible *with $\mathcal{T}$ if, $P =_{\mathcal{RE}} Q$ implies $\mathcal{T} \vdash P \Leftrightarrow Q$,*

$\mathcal{T}$ and $\mathcal{RE}$ are said to be compatible *if $\mathcal{T}$ is compatible with $\mathcal{RE}$ and vice-versa.*

For any class rewrite system $\mathcal{RE}$, we denote $\mathcal{T}_{\mathcal{RE}}$ any theory such that $\mathcal{RE}$ and $\mathcal{T}_{\mathcal{RE}}$ are compatible.

Proposition 1.1 *Let $\mathcal{T}$ be a given theory. If the class rewrite system $\mathcal{RE}$ is compatible with $\mathcal{T}$, then we have:*

$$\mathcal{T}, \Gamma \vdash \Delta \quad \textit{if and only if} \quad \mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta.$$

$$\frac{}{P \vdash_{\mathcal{RE}} Q}\text{axiom} \quad \text{if } P =_{\mathcal{RE}} Q \qquad \frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta \quad \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} \Delta}\text{cut} \quad \text{if } P =_{\mathcal{RE}} Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\text{contr-l} \quad \text{if } P =_{\mathcal{RE}} Q_1 =_{\mathcal{RE}} Q_2 \qquad \frac{\Gamma \vdash_{\mathcal{RE}} Q_1, Q_2, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}\text{contr-r} \quad \text{if } P =_{\mathcal{RE}} Q_1 =_{\mathcal{RE}} Q_2$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\text{weak-l} \qquad \frac{\Gamma \vdash_{\mathcal{RE}} \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}\text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\wedge\text{-l} \quad \text{if } R =_{\mathcal{RE}} (P \wedge Q) \qquad \frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\wedge\text{-r} \quad \text{if } R =_{\mathcal{RE}} (P \wedge Q)$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta \quad \Gamma, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\vee\text{-l} \quad \text{if } R =_{\mathcal{RE}} (P \vee Q) \qquad \frac{\Gamma \vdash_{\mathcal{RE}} P, Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\vee\text{-r} \quad \text{if } R =_{\mathcal{RE}} (P \vee Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \Gamma, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\Rightarrow\text{-l} \quad \text{if } R =_{\mathcal{RE}} (P \Rightarrow Q) \qquad \frac{P \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\Rightarrow\text{-r} \quad \text{if } R =_{\mathcal{RE}} (P \Rightarrow Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\neg\text{-l} \quad \text{if } R =_{\mathcal{RE}} \neg P \qquad \frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\neg\text{-r} \quad \text{if } R =_{\mathcal{RE}} \neg P$$

$$\frac{}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\bot\text{-l} \quad \text{if } P =_{\mathcal{RE}} \bot$$

$$\frac{\Gamma, Q\{x \leftarrow t\} \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}(Q, t)\ \forall\text{-l} \quad \text{if } P =_{\mathcal{RE}} \forall x\ Q \qquad \frac{\Gamma \vdash_{\mathcal{RE}} Q\{x \leftarrow y\}, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}(Q, y)\ \forall\text{-r} \quad \text{if } P =_{\mathcal{RE}} \forall x\ Q$$

$$\frac{\Gamma, Q\{x \leftarrow y\} \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}(Q, y)\ \exists\text{-l} \quad \text{if } P =_{\mathcal{RE}} \exists x\ Q \qquad \frac{\Gamma \vdash_{\mathcal{RE}} Q\{x \leftarrow t\}, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}(Q, t)\ \exists\text{-r} \quad \text{if } P =_{\mathcal{RE}} \exists x\ Q$$

Figure 1: The sequent calculus modulo

Proof: The "only if" part is an obvious induction on the structure of the derivation of $\mathcal{T}, \Gamma \vdash \Delta$, expliciting the witnesses using first-order matching.

The "if" part is an induction on the structure of the derivation of $\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta$.

First notice that using the contraction rule any proof of this statement can be transformed into another where the propositions of $\mathcal{T}$ appear in the left part of every sequent.

As an example, we give the case of the $\wedge$-r rule.

If the proof has the form:

$$\dfrac{\dfrac{\pi}{\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} P, \Delta} \quad \dfrac{\rho}{\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} Q, \Delta}}{\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} R, \Delta}\ \wedge\text{-r} \quad \text{where } R =_{\mathcal{RE}} P \wedge Q$$

By induction hypothesis we have proofs π' and ρ' of $\mathcal{T}, \Gamma \vdash P, \Delta$ and $\mathcal{T}, \Gamma \vdash Q, \Delta$. We build the proof:

$$\dfrac{\dfrac{\pi'}{\mathcal{T}, \Gamma \vdash P, \Delta} \quad \dfrac{\rho'}{\mathcal{T}, \Gamma \vdash Q, \Delta}}{\mathcal{T}, \Gamma \vdash P \wedge Q, \Delta}\ \wedge\text{-r}$$

We have $R =_{\mathcal{RE}} (P \wedge Q)$, thus by hypothesis $\mathcal{T}, \Gamma \vdash (P \wedge Q) \Rightarrow R$. The *modus ponens* is a derived rule of sequent calculus:

$$\dfrac{\dfrac{\dfrac{}{\gamma, B \vdash B, \delta}\ \text{axiom} \quad \dfrac{\dfrac{\dots}{\gamma \vdash A, \delta}}{\gamma \vdash A, B, \delta}\ \text{weak-r}}{\gamma, A \Rightarrow B \vdash B, \delta}\ \Rightarrow\text{-l} \quad \dfrac{\dfrac{\dots}{\gamma \vdash A \Rightarrow B, \delta}}{\gamma \vdash A \Rightarrow B, B, \delta}\ \text{weak-r}}{\gamma \vdash B, \delta}\ \text{cut}$$

Thus we can build a proof of $\mathcal{T}, \Gamma \vdash R, \Delta$. $\square$

$\mathcal{T}$ can be internalized in the sequent calculus modulo, formally:

Corollary 1.1 (Equivalence) *If the theory $\mathcal{T}$ and the class rewrite system $\mathcal{RE}$ are compatible then we have:*

$$\mathcal{T}, \Gamma \vdash \Delta \quad \textit{if and only if} \quad \Gamma \vdash_{\mathcal{RE}} \Delta.$$

Remark 1.4 *If we have only rules of the form $L \to R$ and equations of the for $L = R$ where L and R are propositions, a natural candidate for the theory $\mathcal{T}$ is the universal closure of the propositions $L \Leftrightarrow R$. This way the system $\mathcal{RE}$ is compatible with the theory $\mathcal{T}$ and vice-versa.*

If we have equations of the form $l = r$ where l and r are terms, we can add the universal closure of the propositions $l = r$. But this requires to have an equality predicate in the language and hence the axioms of equality. If we do not have an equality predicate, then we can take the universal closure of all the propositions of the form $P \Leftrightarrow Q$ where $P =_{\mathcal{RE}} Q$.

2 Extended narrowing and resolution

We are now extending the standard resolution based calculus on clausal forms to a calculus where equalities are built-in. For reasons that will appear later, in the rest of the paper, we assume the following properties of the relation $\longrightarrow_{\mathcal{RE}}$:

Assumption 2.1

- *the relation $\longrightarrow_{\mathcal{RE}}$ is confluent and weakly terminating,*
- *the cut rule is redundant in the sequent calculus modulo $\mathcal{RE}$.*

The last property is proved in [DW98] for a large class of systems including the simplification of integral rings and the first-order presentation of higher-order logic introduced above.

Definition 2.1 *A* clause *is a finite set $\{A_1, \ldots, A_n\}$ of propositions, such that every A_i is either an atomic proposition or the negation of an atomic proposition. A clause is said* ground *when it contains no variable. The empty clause is denoted $\Box$.*

Definition 2.2 (Clause form) *Let Φ be a set of propositions and A a proposition, we write Φ, A for the set $\Phi \cup \{A\}$. Let E be a set of sets of propositions and Φ a set of propositions, we write E, Φ for the set $E \cup \{\Phi\}$.*

We consider the following transformations on sets of sets of propositions.

- $E, (\Phi, A \wedge B) \longrightarrow E, (\Phi, A), (\Phi, B)$
- $E, (\Phi, A \vee B) \longrightarrow E, (\Phi, A, B)$
- $E, (\Phi, A \Rightarrow B) \longrightarrow E, (\Phi, \neg A, B)$
- $E, (\Phi, \forall x\ A) \longrightarrow E, (\Phi, A\{x \mapsto y\})$ *where y is a fresh variable*
- $E, (\Phi, \exists x\ A) \longrightarrow E, (\Phi, A\{x \mapsto f(y_1, \ldots, y_n)\})$ *where $y_1, \ldots, y_n$ are the free variables of A*
- $E, (\Phi, \bot) \longrightarrow E, \Phi$
- $E, (\Phi, \neg(A \wedge B)) \longrightarrow E, (\Phi, \neg A, \neg B)$
- $E, (\Phi, \neg(A \vee B)) \longrightarrow E, (\Phi, \neg A), (\Phi, \neg B)$
- $E, (\Phi, \neg(A \Rightarrow B)) \longrightarrow E, (\Phi, A), (\Phi, \neg B)$
- $E, (\Phi, \neg\forall x\ A) \longrightarrow E, (\Phi, \neg A\{x \mapsto f(y_1, \ldots, y_n)\})$ *where $y_1, \ldots, y_n$ are the free variables of A*
- $E, (\Phi, \neg\exists x\ A) \longrightarrow E, (\Phi, \neg A\{x \mapsto y\})$ *where y is a fresh variable*

$$\frac{\{A_1,\dots,A_n,B_1,\dots,B_m\}\,[E_1] \quad \{\neg C_1,\dots,\neg C_p,D_1,\dots,D_q\}\,[E_2]}{\{B_1,\dots,B_m,D_1,\dots,D_q\}\,[E_1 \cup E_2 \cup \{A_1 =^?_{\mathcal{E}} A_2 \dots =^?_{\mathcal{E}} A_n =^?_{\mathcal{E}} C_1 \dots =^?_{\mathcal{E}} C_p\}]}\ \textbf{Resolution}$$

$$\frac{C\,[E]}{c\ell(C[r]_p)\,[E \cup \{C_{|p} =^?_{\mathcal{E}} l\}]}\ \textbf{Narrowing} \quad \text{if } l \to r \in \mathcal{R}$$

Figure 2: Extended narrowing and resolution (ENAR)

- $E, (\Phi, \neg\bot) \longrightarrow E$
- $E, (\Phi, \neg\neg\, A) \longrightarrow E, (\Phi, A)$

This transformation terminates, as the multiset of pairs $< a, b >$ decreases for each rule, where a is the number of occurrences of the symbols $\wedge$, $\vee$, $\Rightarrow$, $\bot$, $\forall$, $\exists$ and b the number of occurrences of the symbol $\neg$ in the sets of propositions elements of E.

The set of sets of propositions obtained this way is a set of clauses.

If E is a set of sets of propositions, we write $c\ell(E)$ its clause form.

Definition 2.3 *For some equational theory $\mathcal{E}$, an equation modulo $\mathcal{E}$ (for short equation) is a pair of terms or atomic propositions denoted $t =^?_{\mathcal{E}} t'$. A substitution σ is an $\mathcal{E}$-solution of $t =^?_{\mathcal{E}} t'$ when $\sigma(t) =_{\mathcal{E}} \sigma(t')$. It is an $\mathcal{E}$-solution of an equation system E when it is a solution of all the equations in E.*

Definition 2.4 *A* constrained clause *(also called a* clause with constraint*) is a pair $C\,[E]$ such that C is a clause and E is a set of equations. It schematizes the set of all instances of C by the solutions of E.*

If C is a set of clauses $\{C_1,\dots,C_n\}$ and E is a set of constraints, then $C\,[E]$ is a notation for the set $\{C_1\,[E],\dots,C_n\,[E]\}$.

This definition could be extended to more general constraints, in particular including ordering constraints on terms, see [KKR90].

Definition 2.5 *Let C be a set of constrained clauses, we write*

$$C \mapsto\!\!\!\!\twoheadrightarrow C'\,[E']$$

if the constrained clause $C'\,[E']$ can be deduced from the clauses of C using finitely many applications of the extended **Narrowing** *and* **Resolution** *rules described in Figure 2.*

Remark 2.1 *As usual, before applying the* **Resolution** *rule, we rename the clauses in such a way that all their free variables are fresh.*

Remark 2.2 *Notice that when $\mathcal{R}$ is empty the rule* **Narrowing** *never applies and we get back equational resolution. When $\mathcal{R}$ and $\mathcal{E}$ are both empty we get back resolution. Notice also that these two inference rules are* not *embedding paramodulation [Pet83] since* **Narrowing** *is always performed with the same built-in equality.*

The main theorem, proved in the next sections, states the correctness and completeness of the ENAR method with respect to the standard as well as equational sequent calculi:

Theorem 2.1 (Main Result) *Let $\mathcal{RE}$ be a decidable class rewrite system such that $\longrightarrow_{\mathcal{RE}}$ is confluent and weakly terminating and the cut rule is redundant in sequent calculus modulo $\mathcal{RE}$. Then, for all propositions $A_1, \dots, A_n, B_1, \dots, B_m$, we have the following equivalences:*

$$\begin{gathered}\mathcal{T}_{\mathcal{RE}}, A_1, \dots, A_n \vdash B_1, \dots, B_m \\ \Leftrightarrow \\ A_1, \dots, A_n \vdash_{\mathcal{RE}} B_1, \dots, B_m \\ \Leftrightarrow \\ c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})\,[\emptyset] \mapsto\!\!\!\twoheadrightarrow \Box\,[E]\end{gathered}$$

where E is an $\mathcal{E}$-unifiable set of equations.

The proof of this fundamental result is detailed in the next sections. The roadmap of the proof construction is the following:

$$\begin{tikzcd}
\mathcal{T}_{\mathcal{RE}}, \Gamma \models \Delta \arrow[rr, bend left, leftrightarrow, "G\ddot{o}del\ 30"] & \Leftrightarrow & \mathcal{T}_{\mathcal{RE}}, \Gamma \vdash \Delta \arrow[rr, bend left, leftrightarrow, "Cor.\ 1.1"] & \Leftrightarrow & \Gamma \vdash_{\mathcal{RE}} \Delta \arrow[rr, bend left, "Prop.\ 3.7"] & \Leftrightarrow & c\ell(\Gamma, \neg\Delta) \hookrightarrow \Box \arrow[ll, bend left, "Prop.\ 3.2"] \arrow[rr, bend left, "Cor.\ 4.1"] & \Leftrightarrow & c\ell(\Gamma, \neg\Delta)\,[\emptyset] \mapsto\!\!\!\twoheadrightarrow \Box\,[E] \arrow[ll, bend left, "Prop.\ 4.1"]
\end{tikzcd}$$

3 An intermediate system

In order to prove Theorem 2.1, we introduce an intermediate system called *Ground Extended Resolution.* Note that the clauses in this system do not need to be ground for applying an inference rule.

Definition 3.1 *Let E be a set of clauses, we write*

$$E \hookrightarrow c$$

if the clause c can be deduced from the clauses of E using finitely many applications of the rules described in Figure 3. This means that there exists a derivation of the clause c under the assumptions E, i.e. a sequence $c_1, \dots, c_n$ such that either $n = 0$ and c is an element of E or $n \geq 1$, $c_n = c$ and each c_i is produced by the application of a rule in Figure 3 from clauses of the set $E \cup \{c_1, \dots, c_{i-1}\}$.

$$\frac{C}{C\{x \mapsto t\}} \text{ Instantiation}$$

$$\frac{C}{C'} \text{ Conversion} \quad \text{if } C =_{\mathcal{E}} C'$$

$$\frac{C}{C'} \text{ Reduction} \quad \text{if } C \to_{\mathcal{R}} P \text{ and } C' \in c\ell(P)$$

$$\frac{\{C, A\} \quad \{C', \neg A\}}{\{C, C'\}} \text{ Ground resolution}$$

Figure 3: Ground extended resolution (GER)

Some proofs by induction on the length of derivations below are destructuring the derivations in a top-down manner. These proofs use the following remark.

Remark 3.1 *If $n \geq 1$ and $c_1, \dots, c_n$ is a derivation of c under E then $c_2, \dots, c_n$ is a derivation of c under $E \cup \{c_1\}$.*

Notation Let $C = \{A_1, \dots, A_n\}$ be a set of propositions. We write $\overline{\forall} C$ for the proposition $\forall x_1 \ \dots \ \forall x_p \ (A_1 \vee \dots \vee A_n)$ where $x_1, \dots, x_p$ are the variables of $A_1, \dots, A_n$, and the proposition $\bot$ if C is empty.

3.1 Soundness

We first start with the well-known proposition on the correctness of the clause form. Notice that the proof of this proposition uses Skolem theorem (i.e. the correctness of skolemization).

Proposition 3.1 (Correctness of the clause form) *Let $A_1, \dots, A_n, B_1, \dots, B_m$ be propositions, and $\mathcal{T}$ be a set of propositions. When:*

$$\{C_1, \dots, C_p\} = c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}),$$

we have:

$$\mathcal{T}, A_1, \dots, A_n \vdash B_1, \dots, B_m \quad \Leftrightarrow \quad \mathcal{T}, \overline{\forall} C_1, \dots, \overline{\forall} C_p \vdash$$

Corollary 3.1 *Let $A_1, \dots, A_n, B_1, \dots, B_m$ be propositions, when:*

$$\{C_1, \dots, C_p\} = c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}),$$

we have:

$$A_1, \dots, A_n \vdash_{\mathcal{RE}} B_1, \dots, B_m \quad \Leftrightarrow \quad \overline{\forall} C_1, \dots, \overline{\forall} C_p \vdash_{\mathcal{RE}}$$

Proposition 3.2 (GER Soundness) *Let $A_1, \ldots, A_n, B_1, \ldots, B_m$ be propositions. If:*

$$c\ell(\{\{A_1\}, \ldots, \{A_n\}, \{\neg B_1\}, \ldots, \{\neg B_m\}\}) \hookrightarrow \Box$$

then:

$$A_1, \ldots, A_n \vdash_{\mathcal{RE}} B_1, \ldots, B_m$$

Proof: Let $\{C_1, \ldots, C_p\} = c\ell(\{\{A_1\}, \ldots, \{A_n\}, \{\neg B_1\}, \ldots, \{\neg B_m\}\})$.
We know that if $\overline{\forall} C_1, \ldots, \overline{\forall} C_p \vdash_{\mathcal{RE}}$ then $A_1, \ldots, A_n \vdash_{\mathcal{RE}} B_1, \ldots, B_m$. Thus, we only need to prove that if $C_1, \ldots, C_n \hookrightarrow \Box$ then $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash_{\mathcal{RE}}$. We prove this by induction on the structure of the derivation of $C_1, \ldots, C_n \hookrightarrow \Box$.

- If the derivation is empty, then one of the clauses C_i is $\Box$. Thus $\overline{\forall} C_1, \ldots, \overline{\forall} C_p \vdash_{\mathcal{RE}}$.
- If the first rule is **Ground resolution** leading to a clause C', then we have $C_i = C_i' \cup \{A\}$, $C_j = C_j' \cup \{\neg A\}$ and $C' = C_i' \cup C_j'$. We have $C_1, \ldots, C_n, C' \hookrightarrow \Box$, thus by induction hypothesis we have a proof of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_p, \overline{\forall} C' \vdash_{\mathcal{RE}}$. The sequent $\overline{\forall} C_i, \overline{\forall} C_j \vdash_{\mathcal{RE}} \overline{\forall} C'$ is provable, thus by the cut rule we get a proof of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_p, \overline{\forall} C_i, \overline{\forall} C_j \vdash_{\mathcal{RE}}$ and by contraction a proof of $\overline{\forall} C_1, \ldots, \overline{\forall} C_p \vdash_{\mathcal{RE}}$.
- If the first rule is **Reduction** reducing the proposition C_i to C_i', then let the set $\{D_1, \ldots, D_p\}$ be $c\ell(\{C_i'\})$. We have $C_1, \ldots, C_n, D_j \hookrightarrow \Box$, thus by induction hypothesis we have a proof of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n, \overline{\forall} D_j \vdash_{\mathcal{RE}}$ and by weakening of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n, \overline{\forall} D_1, \ldots, \overline{\forall} D_p \vdash_{\mathcal{RE}}$.
  The set $\{C_1, \ldots, C_n, D_1, \ldots, D_p\}$ is the clause form of $\{\{\overline{\forall} C_1\}, \ldots, \{\overline{\forall} C_n\}, \{\overline{\forall} C_i'\}\}$, thus by correctness of the clause form the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n, \overline{\forall} C_i' \vdash_{\mathcal{RE}}$ is provable and thus the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n, \overline{\forall} C_i \vdash_{\mathcal{RE}}$ also. By contraction we get a proof of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash_{\mathcal{RE}}$.
- If the first rule is **Conversion** converting the proposition C_i to C_i', then we have $C_1, \ldots, C_n, C_i' \hookrightarrow \Box$, thus by induction hypothesis we have a proof of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n, \overline{\forall} C_i' \vdash_{\mathcal{RE}}$ and by contraction we get a proof of $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash_{\mathcal{RE}}$.
- If the first rule is **Instantiation**, let C_i be the instantiated clause. We have $C_1, C_2, \ldots, C_n, C_i\{x \mapsto t\} \hookrightarrow \Box$, thus by induction hypothesis we have a proof of $\overline{\forall} C_1, \ldots, \overline{\forall} C_n, \overline{\forall}(C_i\{x \mapsto t\}) \vdash_{\mathcal{RE}}$. The sequent $\overline{\forall} C_i \vdash_{\mathcal{RE}} \overline{\forall}(C_i\{x \mapsto t\})$ is provable thus we get by the cut rule a proof of the sequent $\overline{\forall} C_1, \ldots, \overline{\forall} C_p, \overline{\forall} C_i \vdash_{\mathcal{RE}}$ and by contraction a proof of $\overline{\forall} C_1, \ldots, \overline{\forall} C_p \vdash_{\mathcal{RE}}$.

$\Box$

3.2 Completeness

Because we have rewrite rules transforming atomic propositions into non atomic ones, the transformation to clause form (including skolemization) cannot be done as an initial part of the algorithm but it must be done on the fly. Thus if we try to prove that, if $C_1, ... C_n$

are clauses, ($\overline{\forall} C_1, ..., \overline{\forall} C_n \vdash_{\mathcal{RE}}$) implies ($C_1, ..., C_n \hookrightarrow \Box$) the recursion does not go through, because propositions that are not clauses may occur in the proof of $\overline{\forall} C_1, ..., \overline{\forall} C_n \vdash_{\mathcal{RE}}$. Thus we have to prove directly that

$$A_1 \dots A_n \vdash_{\mathcal{RE}} B_1 \dots B_p$$

implies

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_p\}\}) \hookrightarrow \Box$$

The difficult case is for the left rule of the universal quantifier and the right rule of the existential quantifier. Indeed, in these cases, a variable of a proposition is instantiated by a term and thus the set of free variables is modified. So the skolemization of the propositions gives a completely different output.

For instance, if we have a proof of the form:

$$\frac{\exists y \; P(h(u,v,w), y) \vdash \dots}{\forall x \; \exists y \; P(x,y) \vdash \dots} \; \forall l$$

then skolemizing the proposition $\forall x \; \exists y \; P(x,y)$ yields $P(x, f(x))$ with a unary Skolem symbol f, while skolemizing the proposition $\exists y \; P(h(u,v,w), y)$ yields $P(h(u,v,w), g(u,v,w))$ with a ternary Skolem symbol g. In such a case, we have to build a refutation of $P(x, f(x))$ using one of $P(h(u,v,w), g(u,v,w))$. To achieve this goal we need the following propositions, that are related to Skolem theorem and are proved using techniques also used in syntactical proofs of Skolem theorem.

Proposition 3.3 (A form of Skolem theorem for the system $\hookrightarrow$) *Let $w_1, \dots, w_n$ be variables, $t_1, \dots, t_p$ be terms and f and g be two fresh function symbols. From a derivation of:*

$$E \cup c\ell(A\{z \mapsto f(a_1, \dots, a_n)\}) \hookrightarrow \Box$$

we can build one of:

$$E \cup c\ell(A\{z \mapsto g(t_1\{w_1 \mapsto a_1, \dots, w_n \mapsto a_n\}, \dots, t_p\{w_1 \mapsto a_1, \dots, w_n \mapsto a_n\}\})) \hookrightarrow \Box$$

Proof: By induction on the length of the derivation. $\Box$

Proposition 3.4 *From a derivation of $E \cup c\ell(\{C\{x \mapsto t\}\}) \hookrightarrow \Box$ for some term t, we can build one of $E \cup c\ell(\{C\{x \mapsto y\}\}) \hookrightarrow \Box$.*

Proof: We proceed by induction on the number of connectors and quantifiers of C. If x has no occurrence in C then the result is trivial. Otherwise we detail the possible form of the proposition C.

- If the proposition C is atomic then $c\ell(\{C\{x \mapsto y\}\}) = C\{x \mapsto y\}$. With the rule **Instantiation**, we can derive the clause $C\{x \mapsto t\}$ from the clause $C\{x \mapsto y\}$. Hence, from a derivation of $E \cup C\{x \mapsto t\} \hookrightarrow \Box$ we can build one of $E \cup C\{x \mapsto y\} \hookrightarrow \Box$.

- If $C = D_1 \wedge D_2$ then:

$$C\{x \mapsto t\} = D_1\{x \mapsto t\} \wedge D_2\{x \mapsto t\}$$
$$c\ell(\{C\{x \mapsto t\}\}) = \{c\ell(\{D_1\{x \mapsto t\}\}), c\ell(\{D_2\{x \mapsto t\}\})\}$$

Thus, we have a derivation of:

$$E \cup \{c\ell(\{D_1\{x \mapsto t\}\}), c\ell(\{D_2\{x \mapsto t\}\})\} \hookrightarrow \Box$$

by induction hypothesis we have a derivation of:

$$E \cup \{c\ell(\{D_1\{x \mapsto t\}\}), c\ell(\{D_2\{x \mapsto y_2\}\})\} \hookrightarrow \Box$$

and

$$E \cup \{c\ell(\{D_1\{x \mapsto y_1\}\}), c\ell(\{D_2\{x \mapsto y_2\}\})\} \hookrightarrow \Box$$

Substituting y by y_1 and y_2 with the rule **Instantiation**, we get a derivation of:

$$E \cup \{c\ell(\{D_1\{x \mapsto y\}\}), c\ell(\{D_2\{x \mapsto y\}\})\} \hookrightarrow \Box$$

i.e.

$$E \cup \{c\ell(\{C\{x \mapsto y\}\})\} \hookrightarrow \Box$$

- If $C = \exists z\ D$ then:

$$C\{x \mapsto t\} = \exists z\ D\{x \mapsto t\}$$

Let $\{u_1, \ldots, u_n, x\} = FV(C)$ and $\{v_1, \ldots, v_p\} = FV(t) - \{u_1, \ldots, u_n\}$.

$$c\ell(C\{x \mapsto y\}) = c\ell(D\{z \mapsto g(u_1, \ldots, u_n, y)\})$$

$$c\ell(C\{x \mapsto t\}) = c\ell(D\{x \mapsto t\}\{z \mapsto f(u_1, \ldots, u_n, v_1, \ldots, v_p)\})$$

By hypothesis, we have a derivation of:

$$E \cup c\ell(D\{x \mapsto t\}\{z \mapsto f(u_1, \ldots, u_n, v_1, \ldots, v_p)\}) \hookrightarrow \Box$$

By proposition 3.3 we get a derivation of:

$$E \cup c\ell(D\{x \mapsto t\}\{z \mapsto g(u_1, \ldots, u_n, t)\}) \hookrightarrow \Box$$

i.e.

$$E \cup c\ell(D\{x \mapsto y\}\{y \mapsto t\}\{z \mapsto g(u_1, \ldots, u_n, t)\}) \hookrightarrow \Box$$

i.e.

$$E \cup c\ell(\{D\{x \mapsto y\}\{z \mapsto g(u_1, \ldots, u_n, y)\}\{y \mapsto t\}) \hookrightarrow \Box$$

By induction hypothesis we get a derivation of:

$$E \cup c\ell((\{D\{x \mapsto y\}\{z \mapsto g(u_1, \ldots, u_n, y)\} \hookrightarrow \Box$$

i.e.

$$E \cup c\ell(\{C\{x \mapsto y\}\}) \hookrightarrow \Box$$

- the case $C = \neg\forall z\ D$ is similar.
- the other cases are similar to the second one.

$\square$

Proposition 3.5 *If there is a derivation:*

$$c\ell(\{\{C\},\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\}) \hookrightarrow \square$$

and one:

$$c\ell(\{\{D\},\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\}) \hookrightarrow \square$$

then we can build a derivation of:

$$c\ell(\{\{C \vee D\},\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\}) \hookrightarrow \square$$

Proof: Let $E = c\ell(\{\{C\}\})$, $F = c\ell(\{\{D\}\})$, $G = c\ell(\{\{C \vee D\}\})$ and $H = c\ell(\{\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\})$. We check that:

$$c\ell(\{\{C\},\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\}) = E \cup H$$

$$c\ell(\{\{D\},\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\}) = F \cup H$$

$$c\ell(\{\{C \vee D\},\{A_2\},\ldots,\{A_n\},\{\neg B_1\},\ldots,\{\neg B_m\}\}) = G \cup H$$

Thus $E \cup H \hookrightarrow \square$ and $F \cup H \hookrightarrow \square$.

Now $G = \{C_1 \cup C_2 \mid C_1 \in E, C_2 \in F\}$. Consider an arbitrary clause Z in F. We transform the derivation of $E \cup H \hookrightarrow \square$ in such a way that each time we use in $E \cup H \hookrightarrow \square$ a clause X in E we use the clause $X \cup Z$ instead. We get this way either a derivation of $G \cup H \hookrightarrow \square$ or a derivation of $G \cup H \hookrightarrow Z$.

If for some clause Z in F we get a derivation of $G \cup H \hookrightarrow \square$ we are done. Otherwise we get a derivation of $G \cup H \hookrightarrow Z$ for every clause Z of F. Using these derivations and the one of $F \cup H \hookrightarrow \square$ we construct a derivation of $G \cup H \hookrightarrow \square$. $\square$

Proposition 3.6 *If $A =_{\mathcal{RE}} B \wedge C$ then there exists propositions B' and C' such that $A \rightarrow^*_{\mathcal{RE}} B' \wedge C'$, $B =_{\mathcal{RE}} B'$ and $C =_{\mathcal{RE}} C'$.*

If $A =_{\mathcal{RE}} B \vee C$ then there exists propositions B' and C' such that $A \rightarrow^*_{\mathcal{RE}} B' \vee C'$, $B =_{\mathcal{RE}} B'$ and $C =_{\mathcal{RE}} C'$.

If $A =_{\mathcal{RE}} B \Rightarrow C$ then there exists propositions B' and C' such that $A \rightarrow^*_{\mathcal{RE}} B' \Rightarrow C'$, $B =_{\mathcal{RE}} B'$ and $C =_{\mathcal{RE}} C'$.

If $A =_{\mathcal{RE}} \neg B$ then there exists a proposition B' such that $A \rightarrow^*_{\mathcal{RE}} \neg B'$ and $B =_{\mathcal{RE}} B'$.

If $A =_{\mathcal{RE}} \bot$ then $A \rightarrow^*_{\mathcal{RE}} \bot$.

If $A =_{\mathcal{RE}} \forall x\ B$ then there exists a proposition B' such that $A \rightarrow^*_{\mathcal{RE}} \forall x\ B'$ and $B =_{\mathcal{RE}} B'$.

If $A =_{\mathcal{RE}} \exists x\ B$ then there exists a proposition B' such that $A \rightarrow^*_{\mathcal{RE}} \exists x\ B'$ and $B =_{\mathcal{RE}} B'$.

Proof: Because the relation $\rightarrow_{\mathcal{RE}}$ is confluent and the rewrite system rewrites only atomic propositions. $\Box$

Proposition 3.7 (GER Completeness) *Let $A_1, \dots, A_n, B_1, \dots, B_m$ be propositions. If:*

$$A_1, \dots, A_n \vdash_{\mathcal{RE}} B_1, \dots, B_m$$

then:

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

Proof: By induction on the structure of a cut free proof of $A_1, \dots, A_n \vdash_{\mathcal{RE}} B_1, \dots, B_m$.

- If the last rule is $\wedge$-l then one of the A_i's (say A_1) rewrites to $C \wedge D$.
  By induction hypothesis:

  $$c\ell(\{\{C\}, \{D\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

  and using the rules **Conversion** and **Reduction** we can deduce the clauses of this set from the set $c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$. Thus:

  $$c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

- If the last rule is $\vee$-l then of the A_i's (say A_1) rewrites to $C \vee D$. By induction hypothesis:

  $$c\ell(\{\{C\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

  and

  $$c\ell(\{\{D\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

  Using proposition 3.5 we get a derivation of:

  $$c\ell(\{\{C \vee D\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

  Using the rules **Conversion** and **Reduction** we can deduce the clauses of this set from the set $c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$. Thus:

  $$c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

- If the last rule is $\wedge$-r or $\Rightarrow$-l, we proceed as for $\vee$-l.
- If the last rule is $\vee$-r or $\Rightarrow$-r, we proceed as for $\wedge$-l.
- If the last rule is $\neg$-l, $\neg$-r, contr-l or contr-r then the clause form of the antecedent and the succedent of this rule is the same, thus we simply apply the induction hypothesis. If the last rule is weak-l or weak-r then the clause form of the antecedent is a subset of the clause form of the succedent, thus we simply apply the induction hypothesis.

- if the last rule in $\bot$-l, then one of the A_i's (say A_1) rewrites to $\bot$.
  The clause $\Box$ is in the set $c\ell(\{\{\bot\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$ hence:

$$c\ell(\{\{\bot\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

  and using the rules **Conversion** and **Reduction** we can deduce the clauses of this set from the set $c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$. Thus:

$$c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

- If the last rule is $\forall$-l then of the A_i's (say A_1) rewrites to $\forall x\ C$. Call:

$$E = c\ell(\{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$$

  By induction hypothesis we have a derivation of $E \cup c\ell(\{C\{x \mapsto t\}\}) \hookrightarrow \Box$ for some term t. By proposition 3.4 we have a derivation of $E \cup c\ell(\{C\{x \mapsto y\}\}) \hookrightarrow \Box$, i.e. one of $E \cup c\ell(\{\forall x\ C\}) \hookrightarrow \Box$.
  Using the rules **Conversion** and **Reduction** we can deduce the clauses of this set from the set $c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$. Thus:

$$c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

- If the last rule is $\exists$-l then of the A_i's (say A_1) rewrites to $\exists x\ C$. By induction hypothesis, we have a derivation:

$$c\ell(\{\{C\{x \mapsto y\}\}, \{A_2\}, \dots, \{A_n\}, \{B_1\}, \dots, \{B_m\}\}) \hookrightarrow \Box$$

  where y is a fresh variable. The clause form of:

$$\{\{\exists x\ C\}, \{A_2\}, \dots, \{A_n\}, \{B_1\}, \dots, \{B_m\}\}$$

  is the one of:

$$\{\{C\{x \mapsto f(x_1, \dots, x_n)\}\}, \{A_2\}, \dots, \{A_n\}, \{B_1\}, \dots, \{B_m\}$$

  where f is a Skolem symbol.
  In the derivation of:

$$c\ell(\{\{C\{x \mapsto y\}\}, \{A_2\}, \dots, \{A_n\}, \{B_1\}, \dots, \{B_m\}\}) \hookrightarrow \Box$$

  we replace y by $f(x_1, \dots, x_n)$ and we get a derivation of:

$$c\ell(\{\{\exists x\ C\}, \{A_2\}, \dots, \{A_n\}, \{B_1\}, \dots, \{B_m\}\}) \hookrightarrow \Box$$

  Using the rules **Conversion** and **Reduction** we can deduce the clauses of this set from the set $c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})$. Thus:

$$c\ell(\{\{A_1\}, \{A_2\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

- If the last rule is $\forall$-r then we proceed as for $\exists$-l.
- If the last rule is $\exists$-r then we proceed as for $\forall$-l.
- If the last rule is an axiom then $n = p = 1$ and $A_1 =_{\mathcal{RE}} B_1$.
  We build a derivation of $c\ell(\{\{A_1\}, \{\neg B_1\}\}) \hookrightarrow \Box$ with the rules **Conversion**, **Reduction** and **Ground resolution**.

$\Box$

Proposition 3.8 *Let $A_1, \dots, A_n, B_1, \dots, B_m$ be propositions.*

$$A_1, \dots, A_n \vdash_{\mathcal{RE}} B_1, \dots, B_m$$

if and only if

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots \{\neg B_m\}\}) \hookrightarrow \Box$$

Remark 3.2 *We may restrict the rule* **Instantiation** *to $\mathcal{RE}$-normal terms while keeping a complete system.*

4 Proof of the main theorem

4.1 Soundness

Proposition 4.1 (ENAR Soundness) *If $c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})[\emptyset] \mapsto\!\!\!\!\twoheadrightarrow \Box[C]$, where C is a unifiable set of constraints, then $A_1, \dots, A_n \vdash_{\mathcal{RE}} B_1, \dots, B_m$.*

Proof: By induction on the structure of the derivation of

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\})[\emptyset] \mapsto\!\!\!\!\twoheadrightarrow \Box[C]$$

we build a derivation of

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1\}, \dots, \{\neg B_m\}\}) \hookrightarrow \Box$$

Let $\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ be a unifier of C.

- If the first rule is **Resolution**, we substitute with the rule **Instantiation** the variables x_i of the renamed clauses by the corresponding t_i, with the rule **Conversion** we convert these propositions in such a way that they have opposite literals and we apply the rule **Ground resolution**.
- If the first rule is **Narrowing**, we substitute the variables x_i of the clause by the corresponding t_i with the rule **Instantiation** and we reduce it with the rule **Reduction**.

$\Box$

4.2 Completeness

Proposition 4.2 *Consider*

- *a clause $d \cup \{a\}$,*
- *a set C of equations and a substitution θ, solution of C,*
- *a clause d' and propositions $a'_1, \dots, a'_p$ such that $\theta d' =_{\mathcal{E}} d$ and $\forall i, \theta a'_i =_{\mathcal{E}} a$,*
- *a proposition b such that $a \rightarrow_{\mathcal{R}} b$ and a clause $c \in c\ell(d \cup \{b\})$.*

Then, the rule **Narrowing** *can be applied to the constrained clause $(d' \cup \{a'_1, \dots, a'_p\})[C]$ leading, in several steps, to a constrained clause $c'[C']$ and there is a substitution θ', bigger than θ, solution of C', such that $\theta' c' =_{\mathcal{E}} c$.*

Proof: The clause c is an element of $c\ell(d \cup \{b\})$, and $\theta d' =_{\mathcal{E}} d$, thus there exists a clause $c_0 \in c\ell(\theta d' \cup \{b\})$ such that $c_0 =_{\mathcal{E}} c$.

We have $\theta a'_i =_{\mathcal{E}} a$ and $a \rightarrow_{\mathcal{R}} b$ thus, as $\mathcal{R}$ applies only to propositions, we have $\theta a'_i \rightarrow_{\mathcal{R},\mathcal{E}} b$.

Hence the proposition $\theta a'_i$ contains an occurrence u_i such that $(\theta a'_i)_{|u_i} =_{\mathcal{E}} \sigma_i l_i$ for some substitution σ_i and rule $l_i \rightarrow r_i$. The substitution θ is trivially $\mathcal{R}, \mathcal{E}$-normal (because $\mathcal{R}$ reduces only propositions and θ substitutes variables by terms). Hence the occurrence u_i is an occurrence of a'_i and $(\theta a'_i)_{|u_i} = \theta(a'_{i\ |u_i})$.

Let $C' = C \cup \bigcup \{a'_{i\ |u_i} = l_i\}$ and $\theta' = \theta \cup \bigcup \sigma_i$. The substitution θ' is a solution of C' and there is a clause $c' \in c\ell(d' \cup \{a'_i[r_i]_{u_i})$ such that $\theta' c' = c_0$ and hence $\theta' c' =_{\mathcal{E}} c$.

The rule **Narrowing** applies to $d' \cup \{a'_1, \dots, a'_p\}$ leading to the constrained clause $c'[C']$. $\Box$

Proposition 4.3 *If E is a set of constrained clauses, $\hat{E}$ a set that contains renamed copies of clauses of E, θ an $\mathcal{E}$-unifier of all the constraints of $\hat{E}$, F a set of clauses such that $\theta \hat{E} =_{\mathcal{E}} F$ and $F \hookrightarrow \Box$, then $E \mapsto\!\!\!\rightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints.*

Proof: We have $F \hookrightarrow \Box$ hence there exists a derivation $c_1, \dots, c_n$ of the clause $\Box$ under F. We reason by induction on n.

If $n = 0$, the set F contains the empty clause and hence E contains a clause $\Box[C]$ where C is a set of constraints and θ is a solution of C.

If $n \geq 1$ then $c_2, \dots, c_n$ is a derivation of $\Box$ under the set of assumptions $F \cup \{c_1\}$.

The clause c_1 produced by some rule from elements of F. We detail the four cases.

- If the used rule is **Ground resolution**, the set F contains two clauses that contain opposite literals A and $\neg A$. Thus, in E there are two constrained clauses containing respectively literals $A_1, \dots, A_n$ and $\neg B_1, \dots, \neg B_p$ such that $\theta A_i =_{\mathcal{E}}$

$A =_{\mathcal{E}} \theta B_j$. Therefore, the rule **Resolution** applies to E leading to a constrained clauses $c'[C']$, θ is an $\mathcal{E}$-unifier of C' and $\theta c' = c_1$.

The sequence $c_2, \dots, c_n$ is a derivation of $\Box$ under the assumptions F, c_1 and hence, by induction hypothesis $E, c'[C'] \mapsto\!\!\!\twoheadrightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints. Hence $E \mapsto\!\!\!\twoheadrightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints.

- If the used rule is **Instantiation** $x \mapsto t$, then there exists a clause c in F such that $c_1 = c\{x \mapsto t\}$. There exists a constrained clause c' in $\hat{E}$ such that $\theta c' =_{\mathcal{E}} c$. We let c'' be the clause obtained by renaming x into x' in c'. We let $\hat{E}' = \hat{E} \cup \{c''\}$ and $\theta' = \theta \cup \{x' \mapsto t\}$. The substitution θ' is an $\mathcal{E}$-unifier of all the constraints of $\hat{E}'$ and $\theta' \hat{E}' =_{\mathcal{E}} F \cup \{c_1\}$.

  The sequence $c_2, \dots, c_n$ is a derivation of $\Box$ under $F \cup \{c_1\}$ hence, by induction hypothesis $E \mapsto\!\!\!\twoheadrightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints.
- If the used rule is **Conversion**, we have simply $\theta \hat{E} =_{\mathcal{E}} F =_{\mathcal{E}} F \cup \{c_1\}$.

  The sequence $c_2, \dots, c_n$ is a derivation of $\Box$ under $F \cup \{c_1\}$ hence, by induction hypothesis $E \mapsto\!\!\!\twoheadrightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints.
- If the used rule is **Reduction**, then there is in the set F a clause of the form $d \cup \{a\}$, a proposition b such that $a \rightarrow_{\mathcal{R}} b$ and $c_1 \in c\ell(d \cup \{b\})$. Thus, there exists a constrained clause $(d' \cup \{a'_1, \dots, a'_p\})[G]$ in $\mathcal{E}$ such that $\theta d =_{\mathcal{E}} d'$ and $\theta a'_i =_{\mathcal{E}} a$. By the proposition 4.2 the rule **Narrowing** applies to this constrained clause leading to a clause $c'[G']$ and there exists a substitution θ', $\mathcal{E}$-unifier of all the constraints of $E \cup \{c'[G']\}$ and $\theta' E \cup \{c'[G']\} =_{\mathcal{E}} F \cup \{c_1\}$.

  The sequence $c_2, \dots, c_n$ is a derivation of $\Box$ under $F \cup \{c_1\}$ and hence, by induction hypothesis $E, c'[G'] \mapsto\!\!\!\twoheadrightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints. Hence $E \mapsto\!\!\!\twoheadrightarrow \Box[C]$ where C is an $\mathcal{E}$-unifiable set of constraints.

$\Box$

Corollary 4.1 (ENAR Completeness) *If $A_1, \dots, A_n \vdash_{\mathcal{R}\mathcal{E}} B_1, \dots, B_p$, then:*

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1, \dots, \neg B_p\}\})[\emptyset] \mapsto\!\!\!\twoheadrightarrow \Box[C]$$

where C is a $\mathcal{E}$-unifiable set of constraints.

Proof: By the proposition 3.7, if the sequent $A_1, \dots, A_n \vdash_{\mathcal{R}\mathcal{E}} B_1, \dots, B_p$ is provable then:

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1, \dots, \neg B_p\}\}) \hookrightarrow \Box$$

and thus, by the proposition 4.3

$$c\ell(\{\{A_1\}, \dots, \{A_n\}, \{\neg B_1, \dots, \neg B_p\}\})[\emptyset] \mapsto\!\!\!\twoheadrightarrow \Box[C]$$

where C is a $\mathcal{E}$-unifiable set of constraints. $\Box$

5 A Generalization

So far, we have considered a rewrite system $\mathcal{R}$ containing only rules rewriting atomic propositions to arbitrary propositions. We can extend the method above to a rewrite system containing also rules rewriting terms to terms provided the systems verifies the three assumptions below.

The only difference is in the proof of the proposition 4.2 where we have used twice the fact that the system $\mathcal{R}$ reduces only propositions.

First, we have used the fact that as $\mathcal{R}$ reduces only propositions, if $a =_{\mathcal{E}} a'$ and $a \rightarrow_{\mathcal{R}} b$ then $a' \rightarrow_{\mathcal{R},\mathcal{E}} b$. Now we need a strong coherence property between the relation $\rightarrow_{\mathcal{R},\mathcal{E}}$ and $=_{\mathcal{E}}$:

Assumption 5.1 *If $a =_{\mathcal{E}} a'$ and $a \rightarrow_{\mathcal{R},\mathcal{E}} b$ then there exists a term b' such that $a' \rightarrow^{*}_{\mathcal{R},\mathcal{E}} b'$ and $b =_{\mathcal{E}} b'$.*

Then, we have used the following fact. In a proposition of the form θa, if there is a redex at the occurrence u, then it is a proposition occurrence and thus an occurrence of a.

To have the same property, we now need the substitution θ to be $\mathcal{R},\mathcal{E}$-normal. To maintain this normality hypothesis, we need to use the **Instantiation** rule with $\mathcal{R}\mathcal{E}$-normal, and hence $\mathcal{R},\mathcal{E}$-normal, terms (see remark 3.2) and to apply the rule **Reduction** on innermost $\mathcal{R}$-redexes only. Then to achieve completeness we must prove that any $\hookrightarrow$-deduction can be transformed into another one where we apply the **Reduction** to innermost $\mathcal{R}$-redexes only. This is possible, for instance, when the innermost $\mathcal{R},\mathcal{E}$-reduction terminates.

Assumption 5.2 *The innermost $\mathcal{R},\mathcal{E}$-reduction terminates.*

At last we need to show that innermostness is preserved by the relation $=_{\mathcal{E}}$. We denote $\rightarrow^{i}_{\mathcal{R},\mathcal{E}}$ the reduction of an innermost redex.

Assumption 5.3 *If $a =_{\mathcal{E}} a'$ and $a \rightarrow^{i}_{\mathcal{R},\mathcal{E}} b$ then there exists a term b' such that $a' \rightarrow^{i}_{\mathcal{R},\mathcal{E}} b'$ and $b =_{\mathcal{E}} b'$.*

Under these assumptions the method above is complete.

The assumptions above are similar to those necessary to establish the completeness of equational narrowing modulo [Kir85] which is indeed an instance of the method developed here.

Conclusion

In this paper, we have presented a sequent calculus that operates in the quotient of the set of propositions modulo a congruence which can equate atomic propositions with non atomic ones.

We have given a proof search method based on extended narrowing and resolution and proved that it is sound and complete with respect to the sequent calculus modulo for a large class of systems $\mathcal{RE}$.

When we apply this method to the first-order presentation of higher-order logic above, the rule **Narrowing** specializes exactly to the rule **Splitting** of higher-order resolution [Hue72, Hue73]. The only difference with higher-order resolution is that we are using the combinators S and K and not λ-calculus. Using combinators let skolemization be simpler but let unification be only equational unification modulo the axioms S and K and not modulo the axioms β and η as in higher-order resolution. We believe that a first-order presentation of higher-order logic based not on combinators but on explicit substitutions would simulate exactly higher-order resolution. A first step towards such a result, the expression of higher-order unification as equational unification in the calculus of explicit substitutions, has been achieved in [DHK95]. Giving a full first-order presentation of higher-order logic using explicit substitution is work in progress.

We hope that this method will also be useful for theories stronger than HOL in particular for HOL extended with equalities (e.g. associativity and commutativity of some operations) simulating this way equational higher-order resolution.

References

[And71] P. B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36:414–432, 1971.

[Bac87] L. Bachmair. *Proof methods for equational theories.* PhD thesis, University of Illinois, Urbana-Champaign, (Ill., USA), 1987. Revised version, August 1988.

[BGLS95] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[DHK95] G. Dowek, T. Hardin, and C. Kirchner. Higher-order unification via explicit substitutions, extended abstract. In D. Kozen, editor, *Proceedings of LICS'95*, pages 366–374, San Diego, June 1995.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 6, pages 244–320. Elsevier Science Publishers B. V. (North-Holland), 1990.

[Dow97] G. Dowek. Proof normalization for a first-order formulation of higher-order logic. In E. Gunter and A. Felty, editors, *Proceedinds of* Theorem proving in higher order logics 1997, volume 1275 of *Lecture Notes in Computer Science*, pages 105–119. Springer-Verlag, 1997. Technical Report 3383 INRIA, 1998.

[DW98] G. Dowek and B. Werner. Proof normalization modulo. *Manuscript*, 1998.

[Gal86] J. H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*, volume 5 of *Computer Science and Technology Series*. Harper & Row, New York, 1986.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[Han94] M. Hanus. The integration of functions into logic programming: From theory to practice. *Journal of Logic Programming*, 19&20:583–628, 1994.

[HKLR92] J. Hsiang, H. Kirchner, P. Lescanne, and M. Rusinowitch. The term rewriting approach to automated theorem proving. *Journal of Logic Programming*, 14(1&2):71–99, October 1992.

[HR86] J. Hsiang and M. Rusinowitch. A new method for establishing refutational completeness in theorem proving. In J. Siekmann, editor, *Proceedings 8th International Conference on Automated Deduction, Oxford (UK)*, volume 230 of *Lecture Notes in Computer Science*, pages 141–152. Springer-Verlag, 1986.

[Hue72] G. Huet. *Constrained Resolution: A Complete Method for Higher Order Logic.* PhD thesis, Case Western Reserve University, 1972.

[Hue73] G. Huet. A mechanization of type theory. In *Proceeding of the third international joint conference on artificial intelligence*, pages 139–146, 1973.

[Hue75] G. Huet. A unification algorithm for typed lambda calculus. *Theoretical Computer Science*, 1(1):27–57, 1975.

[Hul80] J.-M. Hullot. Canonical forms and unification. In *Proceedings 5th International Conference on Automated Deduction, Les Arcs (France)*, pages 318–334, July 1980.

[JK86] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15(4):1155–1194, 1986. Preliminary version in Proceedings 11th ACM Symposium on Principles of Programming Languages, Salt Lake City (USA), 1984.

[JK91] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: a rule-based survey of unification. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 8, pages 257–321. The MIT press, Cambridge (MA, USA), 1991.

[JL87] J. Jaffar and J.-L. Lassez. Constraint logic programming. In *Proceedings of the 14th Annual ACM Symposium on Principles Of Programming Languages, Munich (Germany)*, pages 111–119, 1987.

[KB70] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[Kir85] C. Kirchner. *Méthodes et outils de conception systématique d'algorithmes d'unification dans les théories équationnelles.* Thèse de Doctorat d'Etat, Université Henri Poincaré – Nancy 1, 1985.

[KKR90] C. Kirchner, H. Kirchner, and M. Rusinowitch. Deduction with symbolic constraints. *Revue d'Intelligence Artificielle*, 4(3):9–52, 1990. Special issue on Automatic Deduction.

[Kol96] G. Kolata. With major math proof, brute computers show flash of reasoning power. *The New York Times*, 1996. Tuesday December 10.

[KvOvR93] J. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems: introduction and survey. *Theoretical Computer Science*, 121:279–308, 1993.

[LP94] S.-J. Lee and D. Plaisted. Use of replace rules in theorem proving. *Methods of Logic in Computer Science*, 1(2):217–240, 1994.

[McC97] W. McCune. Solution of the robbins problem. *Journal of Automated Reasoning*, 19(3):263–276, 1997.

[Pet83] G. Peterson. A technique for establishing completeness results in theorem proving with equality. *SIAM Journal of Computing*, 12(1):82–100, 1983.

[Plo72] G. Plotkin. Building-in equational theories. *Machine Intelligence*, 7:73–90, 1972.

[PS81] G. Peterson and M. E. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28:233–264, 1981.

[Sti85] M. Stickel. Automated deduction by theory resolution. *Journal of Automated Reasoning*, 1(4):285–289, 1985.

[Vir95] P. Viry. Rewriting modulo a rewrite system. Technical report TR-20/95, Dipartimento di informatica, Università di Pisa, December 1995.

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399