A Model-based Completeness Proof of Extended Narrowing And Resolution

Jürgen Stuber

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Model-based Completeness Proof of Extended Narrowing And Resolution

Jürgen Stuber

N° 4135

March 2001

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4135--FR+ENG

A Model-based Completeness Proof of Extended Narrowing And Resolution

Jürgen Stuber*

Thème 2 — Génie logiciel
et calcul symbolique
Projet PROTHEO

Rapport de recherche n° 4135 — March 2001 — 18 pages

Abstract: We give a proof of refutational completeness for Extended Narrowing And Resolution (ENAR), a calculus introduced by Dowek, Hardin and Kirchner in the context of Theorem Proving Modulo. ENAR integrates narrowing with respect to a set of rewrite rules on propositions into automated first-order theorem proving by resolution. Our proof allows to impose ordering restrictions on ENAR and provides general redundancy criteria, which are crucial for finding nontrivial proofs. On the other hand, it requires confluence and termination of the rewrite system, and in addition the existence of a well-founded ordering on propositions that is compatible with rewriting, compatible with ground inferences, total on ground clauses, and has some additional technical properties. Such orderings exist for hierarchical definitions of predicates. As an exampe we provide such an ordering for a fragment of set theory.

Key-words: automated reasoning, automated theorem proving, resolution, rewriting, narrowing, skolemization, constraints

* stuber@loria.fr http://www.loria.fr/~stuber/

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Preuve de complétude de ENAR par construction de modèle

Résumé : Nous donnons une preuve de complétude refutationelle pour le calcul ENAR, introduit par Dowek, Hardin et Kirchner dans le cadre de la déduction modulo. ENAR (Extended Narrowing And Resolution) intègre la surréduction par rapport à une ensemble de règles de réécriture sur des propositions dans la resolution du premier ordre. Notre preuve permet d'obtenir des restrictions d'ordre sur ENAR et fournit un critère générale de redondance, ceci permettant de trouver des preuves plus grandes. D'un autre coté, la preuve necessite la confluence et la terminaison du système de réécriture, et l'existence d'un ordre bien fondé sur les propositions, compatible avec la réécriture et des inférences closes, totale sur des clauses closes, et possedant en plus quelques proprietés techniques. De tels ordres existent pour des definitions hiérarchiques des predicates. Comme exemple, nous définissons un tel ordre pour un fragment de la théorie des ensembles.

Mots-clés : raisonnement automatique, preuve automatique, résolution, réécriture, surréduction, skolemization, contraintes

1 Introduction

Dowek, Hardin and Kirchner [6] introduce Theorem Proving Modulo and in that context the calculus Extended Narrowing And Resolution (ENAR). They show completeness of ENAR by transforming proofs in a sequent calculus modulo a congruence on formulas into ENAR proofs with respect to the same congruence represented by a term rewriting system, using cut elimination for the sequent calculus in the process. Dowek and Werner [7] show the cut elimination property for the cases of HOL-$\lambda\sigma$, quantifier-free theories and positive theories.

Here we give an alternate completeness proof based on the reduction-of-counterexamples method developed over recent years [3]. This allows to impose ordering restrictions on the calculus and provides a strong notion of redundancy, which is crucial for solving larger problems. The proof requires a well-founded ordering on propositions with certain properties such as compatibility with the rewrite relation. Such orderings exist for hierarchical definitions of predicates. As an exampe we define such an ordering for a small fragment of set theory.

From the viewpoint of automated theorem proving it is interesting to study how the technique for proving refutational completeness can be extended to handle skolemization or even quantifiers in formulas. Since logical equivalence is lost by skolemization, we have to adapt the notion of soundness of the calculus accordingly. By capturing the effect of skolemization in the addition of *skolemization axioms*, we can keep logical equivalence for most of the proof.

Finally, it is interesting to study calculi with built-in theories in order to improve the efficiency of automated theorem provers. It is generally recognized that automated theorem provers have problems proving theorems in theories with permutative axioms like associativity, commutativity, distributivity and the inverse law which are common in algebra, and there have been various approaches to the integration of these axioms into provers [18, 11, 17, 5, 2, 8, 13, 14, 16]. A similar argument holds for the use of equivalences on the level of logical formulas. State-of-the-art resolution theorem provers such as SPASS do a clause normal form transformation once at the beginning, which destroys in particular the equivalences. With some effort it is possible to reconstruct the equivalences [12] and take advantage of them, but to us it seems more fruitful to work towards using them directly. There has been some work on nonclausal resolution by Bachmair and Ganzinger [1], but this does not cover formulas with quantifiers.

2 Preliminaries

We consider first-order logic without equality with respect to fixed sets $\mathcal{P}$ of predicate symbols and $\mathcal{F}$ of function symbols. We assume that $\mathcal{F}$ contains countably many function symbols of each arity, in order to provide sufficiently many fresh function symbols for skolemization. An *atom* is a formula $p(t_1, \ldots, t_n)$ where $p \in \mathcal{P}$ and $t_1, \ldots, t_n$ are terms. Propositions are built from atoms, $\top$ (truth), $\bot$ (falsity), by the junctors $\wedge$, $\vee$, $\neg$, $\rightarrow$ (implication), $\leftrightarrow$

(equivalence), and the quantifiers $\forall$ and $\exists$. We use the double arrow for rewriting, $\Rightarrow$ for a rule or a single step and $\overset{*}{\Rightarrow}$ for the reflexive-transitive closure of $\Rightarrow$. $\overset{!}{\Rightarrow}$ rewrites to normal form, i.e. $s \overset{!}{\Rightarrow} t$ if $s \overset{*}{\Rightarrow} t$ and t is irreducible. We write $P|_\pi$ for the subproposition or subterm of P at the position π, and $P[Q]_\pi$ or $P[t]_\pi$ for the proposition P where we have replaced the subformula or subterm at position π by Q or t, respectively.

A *literal* is either an atom or the negation of an atom, and a *clause* is a disjunction of literals. We will use *constrained clauses* of the form $C\,[\mathcal{C}]$ where C is a clause and $\mathcal{C}$ is a *constraint*. A *syntactic equality constraint* $s \approx t$ is *satisfied* for those ground substitutions σ that unify s and t, i.e. where $s\sigma = t\sigma$. Analogously, for a fixed given ordering $\succ$ on ground terms, σ satisfies an ordering constraint $s \succ t$ if $s\sigma \succ t\sigma$. We will use contraints that are conjunctions of these atomic constraints. The meaning of a constrained clause is the set of ground instances obtained by susbtitutions that satisfy the constraint.

A (finite) *multiset* M over a set S is a function from S into the natural numbers such that $M(x) > 0$ only for finitely many x in S. For each x in S, $M(x)$ denotes the number of occurrences of x in M. The *multiset extension* $\succ_{mul}$ of a strict partial ordering $\succ$ is the strict partial ordering on multisets over S that is defined by $M \succ_{mul} N$ if and only if $M \neq N$ and for all x in S such that $N(x) > M(x)$ there exists an y in S such that $y \succ x$ and $M(y) > N(y)$. We will use that the multiset extension of a total ordering is total, that the multiset extension preserves well-foundedness, and that the ordering on multisets is dominated by the ordering on their maximal elements.

We consider propositions to be modulo associativity and commutativity (AC) for $\vee$ and $\wedge$. In particular, clauses that differ only in the order of their literals are identical. We write $\{t_1/x_1, \ldots, t_n/x_n\}$ for the substitution that replaces x_i by t_i for $i \in \{1, \ldots, n\}$.

We use the following rules for the transformation to clause normal form :

$$\neg\bot \Rightarrow \top \tag{1}$$
$$\neg\top \Rightarrow \bot \tag{2}$$
$$\bot \wedge P \Rightarrow \bot \tag{3}$$
$$\top \wedge P \Rightarrow P \tag{4}$$
$$\bot \vee P \Rightarrow P \tag{5}$$
$$\top \vee P \Rightarrow \top \tag{6}$$
$$P \leftrightarrow Q \Rightarrow (P \rightarrow Q) \wedge (Q \rightarrow P) \tag{7}$$
$$P \rightarrow Q \Rightarrow \neg P \vee Q \tag{8}$$
$$\neg\neg P \Rightarrow P \tag{9}$$
$$\neg(P \vee Q) \Rightarrow \neg P \wedge \neg Q \tag{10}$$
$$\neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q \tag{11}$$
$$\neg(\forall x\, P) \Rightarrow \exists x\, \neg P \tag{12}$$
$$\neg(\exists x\, P) \Rightarrow \forall x\, \neg P \tag{13}$$
$$P \vee (Q_1 \wedge Q_2) \Rightarrow (P \vee Q_1) \wedge (P \vee Q_2) \tag{14}$$

$$\forall x\, P \Rightarrow P\{z/x\} \tag{15}$$
$$\exists x\, P \Rightarrow P\{f(y_1, \ldots, y_n)/x\} \tag{16}$$

where in (15) z is a new variable, and in (16) f is a fresh function symbol and $x, y_1, \ldots, y_n$ are the free variables of P. A *clause normal form* of a proposition P is obtained by exhaustively applying these rules, with the restriction that the rules (1)–(13) must be applied before (15) and (16), in order to apply the quantifier rules only below positive contexts. The clause normal form transformation is nondeterministic by the choice of new variables and fresh function symbols. This is not a problem, as in any context where a clause normal form is needed any one will do, i.e. this is don't-care-nondeterminism. Note that by this definition clauses are not sets, but are formed form the same symbols as logical propositions. In particular, the empty clause is $\bot$. Also, by equivalence modulo AC these clauses behave as multisets, where the same element may occur several times.

If we consider free variables to be universally quantified then (1)–(15) are logical equivalences, while (16) is only an implication from right to left. It becomes an equivalence if we add the implication in the other direction. Thus, we call

$$\forall y_1, \ldots, y_n.(\exists x.P) \rightarrow P\{f(y_1, \ldots, y_n)/x\}.$$

the *skolem axiom* for the *skolem function symbol* f with respect to $\exists x.P$. We call a set of skolem axioms S *fresh* with respect to a set of propositions N if no skolem function symbol of S occurs in N, and there is only one skolem axiom for every skolem function symbol in S. A fresh set of skolem axioms is always obtained when fresh function symbols are used for skolemization.

Lemma 1 *Let N be a set of propositions and S a set of skolem axioms that is fresh with respect to N, and let I be a model of N. Then there exists a model I' of $N \cup S$.*

Proof: Sketch : define the interpretation of skolem functions in I' so that they provide witnesses for the true instances of their corresponding existential formula. Since S is fresh this is possible without changing the truth value of N.

Lemma 1 isolates the argument that the clause normal form transformation preserves satisfiability. By adding skolem axioms to our theory at the beginning, we get logical equivalence for all later steps. This simplifies our arguments below.

3 Rewriting on propositions

Let R_p be a set of rewrite rules on first-order propositions such that left-hand sides are atomic, let R_t be a set of rewrite rules on terms, and let $R = R_p \cup R_t$. The right-hand side of a rule in R may contain only free variables that also occur in the left-hand side. We write T_{R_p} for the logical meaning of R_p, which is the set of logical equivalences $\{l \leftrightarrow r \mid l \Rightarrow r \in R\}$. The intended meaning of the rules in R_t is equality. However, equality is not directly available to us, since we use first-order logic without built-in equality. As an alternative, we may apply

Leibniz' equality to atomic propositions to obtain a set of equivalences that capture the logical meaning of R_t. That is, we let

$$T_{R_t} = \{A[l]_\pi \leftrightarrow A[r]_\pi \mid l \Rightarrow r \in R,\ A \text{ an atom, and } \pi \text{ a position in } A\}.$$

This is adequate, since any model that satisfies T_{R_t} can be factored through the congruence induced by R_t to obtain a model of R_t with respect to first-order logic with equality, while preserving the truth value of propositions. Finally, we let $T_R = T_{R_p} \cup T_{R_t}$. The theory T_R is compatible with the rewrite rules in R in the sense of Dowek, Hardin and Kirchner [6]. It is somewhat smaller than the one given there, as it includes equivalences only for single rewrite steps and relies on the properties of logical equivalence for reflexive-transitive closure and for closure under contexts of logical operators and substitutions.

We assume that R is confluent and terminating modulo AC for $\wedge$ and $\vee$, and write $R(P)$ for the normal form of a proposition P with respect to $\Rightarrow_R$.

4 The inference system ENAR

An *inference system* is a set of inferences on constrained clauses. Each *inference* has a *main premise* C, zero or more *side premises* $C_1, \ldots, C_n$, and a *conclusion* D, and is written

$$\frac{C \qquad C_1 \quad \ldots \quad C_n}{D}.$$

The main premise and the side premises have different roles in the completeness proof and in the resulting notion of redundancy for inferences.

The calculus of Extended Narrowing And Resolution (ENAR) consists of the following two rules operating on constrained clauses :

Extended Resolution $$\frac{\neg A_1 \vee \ldots \vee \neg A_n \vee C\,[\mathcal{C}_1] \qquad B_1 \vee \ldots \vee B_m \vee D\,[\mathcal{C}_2]}{C \vee D\,[\mathcal{C}_1 \wedge \mathcal{C}_2 \wedge A_1 \approx \ldots \approx A_n \approx B_1 \approx \ldots \approx B_m]}$$

where $A_1 \approx \ldots \approx A_n \approx B_1 \approx \ldots \approx B_m$ is an abbreviation for $A_1 \approx A_2 \wedge \ldots \wedge A_1 \approx A_n \wedge A_1 \approx B_1 \wedge \ldots \wedge A_1 \approx B_m$. The main premise of Extended Resolution is $\neg A_1 \vee \ldots \vee \neg A_n \vee C\,[\mathcal{C}_1]$.

Extended Narrowing $$\frac{U\,[\mathcal{C}]}{\mathrm{cl}(U[r]_\pi)\,[\mathcal{C} \wedge (U|_\pi \approx l)]}$$

where $l \Rightarrow r$ is a rule in R and $U|_\pi$ is not a variable.

Here $\mathrm{cl}(P)$ denotes one of the clauses in a clause normal form of P.

This calculus is slightly different from the original one [6], as it doesn't use equality modulo a congruence. On the other hand, we allow the use of rewrite rules on terms.

5 The ordering on propositions

We say that an ordering has the *multiset property* for $\vee$ if $L \succ L'$ for any literal L' in a clause C implies $L \succ C$. We assume an ordering $\succ$ on propositions that is well-founded,

total on ground clauses, that has the multiset property for $\vee$, that satisfies $\neg A \succ A$ for any atom A, that is compatible with rewriting, i.e., $(\Rightarrow_R) \subseteq (\succ)$, and $A \Rightarrow_R P$ implies $A \succ B$ for every ground instance B of an atom in P.

The latter implies compatibility with Extended Narrowing, i.e. $C \succ D$ for every ground inference with main premise C and conclusion D. Compatibility with Extended Resolution follows by the multiset property. Note that we have replaced compatibility with rewrite rules and with contexts by the somewhat weaker compatibility with rewriting, i.e. the application of rewrite rules under contexts. We have done this because it is difficult to obtain compatibility with contexts for quantifiers under negative contexts.

The property that A is greater than any ground instance of some B with free variables is not satisfied by typical term orderings, in particular not by simplification orderings. The separation of propositions and terms can be used to avoid this problem by giving predicate symbols precedence over terms. This technique is applicable in particular for hierarchical definitions of predicates by equivalences, e.g. in set theory. We present an example below.

6 Constructing Herbrand Models that Satisfy the Equivalences

We now define a function closure_R that maps a Herbrand interpretation H_i for ground atoms that are irreducible by R to a Herbrand interpretation for all ground atoms. The mapping is defined in such a way that the interpretation of irreducible atoms is not changed and T_R becomes true in $\text{closure}_R(H_i)$.

Let H_i be a set of ground atoms irreducible by R. We construct a tree from each closed proposition whose inner nodes are labeled by $\wedge$, $\vee$ and $\neg$ and whose leaves are irreducible ground atoms. $\leftrightarrow$ and $\rightarrow$ are always expanded using rules (7) and (8).

1. If P is reducible by R then the tree for P is the tree for the normal form of P with respect to R.
2. The tree for an irreducible ground atom A is a leaf labeled A.
3. The tree for an irreducible proposition $P \wedge Q$ is labeled $\wedge$ at the root and has as children the trees for P and Q.
4. The tree for an irreducible proposition $P \vee Q$ is labeled $\vee$ at the root and has as children the trees for P and Q.
5. The tree for an irreducible proposition $\neg P$ is labeled $\neg$ at the root and has the tree for P as the only child.
6. The tree for an irreducible proposition $\forall x.P$ is labeled $\wedge$ at the root and has as children all trees for $P\{t/x\}$ where t is a ground term.
7. The tree for an irreducible proposition $\exists x.P$ is labeled $\vee$ at the root and has as children all trees for $P\{t/x\}$ where t is a ground term.

Lemma 2 *All the branches of the tree are finite.*

Proof: The only possible source of nontermination is the interaction of rewriting and instantiation of quantifiers. One of the properties of $\succ$ is that a rewrite step followed by instantiation decreases all propositions in the ordering. Since $\succ$ is a well-founded ordering this implies termination. □

Now we label the nodes of the tree with truth values true or false from the bottom up. A leaf A is labeled true if $A \in H_i$, and false otherwise. A node labeled with $\wedge$ is labeled true if all of its children are labeled true, and false otherwise. A node labeled with $\vee$ is labeled true if some of its children is labeled true, and false otherwise. A node labeled $\neg$ is labeled true if its child is labeled false and vice-versa. Since all branches are finite all the nodes are labeled. We let $A \in \text{closure}_R(H_i)$ if the root of the tree for A is labeled true.

Lemma 3 *Let T be a tree for a closed proposition P. Then the root of T is labeled true if and only if P is true in $\text{closure}_R(H_i)$.*

Proof: For atoms this is immediate by the definition of truth in the Herbrand interpretation $\text{closure}_R(H_i)$. For other propositions it is a straightforward structural induction. □

Lemma 4 *$\text{closure}_R(H_i) \models T_R$.*

Proof: Consider some equivalence $A \leftrightarrow P$ in T_R. The tree for A is identical to the tree for P, because both are the tree for the normal form of A with respect to R, which is unique by confluence and termination of R. Hence their truth value is equal and the equivalence holds. □

7 Refutational completeness

To be able to do lift narrowing steps on terms to constrained narrowing we use the standard technique that considers only reduced ground instances on the ground level. This technique was originally used to show completeness of basic narrowing [9], and later for constrained or basic first-order calculi [4, 10]. Formally, a substitution σ is *reduced* if $x\sigma$ is irreducible with respect to R_t for all variables x. An instance is called *reduced* if it is obtained by a reduced substitution. We write $\text{gnd}(N)$ for the set of ground instances of clauses in N, and $\text{rgnd}_R(N)$ for the subset of reduced ground instances of clauses in N. Note that ground instances have to satisfy the constraint. We also consider reduced ground inferences. Since premises can always be made ground, the restriction to reduced instances of inferences restricts only the instantiation of newly introduced variables in conclusions. We will show that ENAR is refutationally complete by showing that ENAR has the reduction property for counterexamples, using the approach of Bachmair and Ganzinger [3].

We will define mutually recursive functions I and P that map a set of ground clauses to a set of ground atoms. Here I stands for "interpretation" and P for "produced". Let M be a set of ground clauses. $P(M)$ defines the interpretation of ground atoms that are irreducible with respect to R, and $I(M)$ extends $P(M)$ to all ground atoms, using the function closure_R defined above.

The definition is with respect to the well-ordering $\succ$ on ground clauses, considering the ground clauses in M in turn. For some ground clause C we write $M^{\prec C}$ ($M^{\preceq C}$) for the ground clauses in M that are smaller than C (less or equal to C).

$$P(M) = \bigcup_{C \in M} \Delta(M, C)$$

$$\Delta(M,C) = \begin{cases} \{A\} & \text{if (i) } C \text{ is false in } I(M^{\prec C}), \\ & \quad \text{(ii) } C = C' \vee A \vee \ldots \vee A, \\ & \quad \text{(iii) } A \succ C', \text{ and} \\ & \quad \text{(iv) } A \text{ is irreducible by } R\text{; or} \\ \emptyset & \text{otherwise.} \end{cases}$$

$$I(M) = \text{closure}_R(P(M))$$

$$I(N) = I(\text{rgnd}_R(N))$$

We write N_C for $\text{rgnd}_R(N)^{\prec C}$. Note that $\Delta(M,C) = \Delta(M^{\prec C}) = \Delta(N_C, C)$, since this is the part of M that is used recursively. Since we start the definition of $I(N)$ with the set of reduced ground instances of N, the set $\Delta(\text{rgnd}_R(N), C)$ is the increment that takes us from $P(\text{rgnd}_R(N)^{\prec C})$ to $P(\text{rgnd}_R(N)^{\preceq C})$.

We say that a ground clause C *produces* A if $A \in \Delta(\text{rgnd}_R(N), C)$. We say that a ground clause C is a *counterexample* for $I(N)$ if it is in $\text{rgnd}_R(N)$ and false in $I(N)$. Let C be the least counterexample for $I(N)$ and let $\mathcal{I}$ be a ground inference with main premise C and conclusion D such that $C \succ D$. We say that $\mathcal{I}$ *reduces the counterexample* C (with respect to $I(N)$) if $I(N) \models \neg D$. An inference system Calc has the *reduction property for counterexamples* (with respect to I) if there is a reduced ground instance of an inference in Calc that reduces C with respect to $I(N)$ for any set N of ground clauses such that $I(N)$ has the least counterexample $C \neq \bot$.

Lemma 5 ENAR *has the reduction property for counterexamples.*

Proof: Let N be a set of clauses, let C be the least counterexample in $I(N)$ and suppose $C \neq \bot$. Then C is the reduced ground instance of a clause $\hat{C}$ in N and has the form $C' \vee L \vee \ldots \vee L$ where $L \succ C'$ for some literal L.

(1) Suppose L is reducible by R. We have $L = A$ or $L = \neg A$ and A is reducible by some rule $B \Rightarrow \hat{P}$ in R, where $A = B\sigma$ and $P = \hat{P}\sigma$. We consider the single R-step $A \Rightarrow P$ applied to $C[A]_\pi$, resulting in the formula $C[P]_\pi$ that is false in $I(N)$, since the equivalence $A \leftrightarrow P$ holds in $I(N)$. Any clause normal form of $C[P]_\pi$ implies $C[P]_\pi$, as skolemization is an implication in the reverse direction. Since $C[P]_\pi$ is false in $I(N)$, the clause normal form is also false in $I(N)$, and there is a ground instance of a clause in the CNF that is false in $I(N)$. As C is a reduced ground instance of some clause $\hat{C}$ in N, the position π can not be a variable position of $\hat{C}$. Hence there exists an Extended Narrowing inference

$$\frac{\hat{C}[\hat{A}]_\pi\,[\mathcal{C}]}{\hat{D}_i\,[\mathcal{C} \wedge \hat{A} \approx B]}$$

such that $\hat{D}_1 \wedge \ldots \wedge \hat{D}_n$ is a clause normal form of $\hat{C}[\hat{P}]$ and $\hat{D}_i\,[\mathcal{C} \wedge \hat{A} \approx B]$ has a ground instance D that is false in $I(N)$ for some $i \in \{1, \ldots, n\}$. Since newly introduced variables are unconstrained we may even choose D to be a reduced ground instance of $\hat{D}_i$. Hence the inference above reduces the counterexample C.

(2) Otherwise L is irreducible by R.

(2.1) Suppose L is positive, i.e. $L = A$ for some ground atom A. Since C is false in $I(N)$, A is false in $I(N)$ and in $P(N)$. All other literals are smaller than A, hence their truth value depends only on the truth values of irreducible atoms smaller than A, which are the same in $I(N)$ and in $I(N_C)$. Therefore C is false in $I(N_C)$. Hence C produces A, $A \in P(N)$ and C is true in $I(N)$, a contradiction.

(2.2) Otherwise L is negative, that is, $L = \neg A$ for some atom A. Since L is false, A is true in $I(N)$, and by irreducibility A must be in $P(N)$. This A is produced by some reduced ground instance D of a clause $\hat{D}$ in N with $C \succ D$. Then $D = D' \vee A \vee \ldots \vee A$, $A \succ D'$, and D' is false in $I(N_D)$. No clause greater than or equal to D can make a literal in D' become true, hence D' is false in $I(N)$. We can resolve C and D :

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

This is a reduced ground instance of Extended Resolution that reduces C. □

Lemma 6 *Let N be a set of clauses that is closed under* ENAR. *Then either $\bot \in N$ or $I(N) \models \mathrm{rgnd}_R(N)$.*

Proof: Suppose $C \neq \bot$ is the least counterexample in N for $I(N)$. Then by the reduction property there exists a reduced ground inference

$$\frac{C \qquad C_1 \quad \ldots \quad C_n}{D}$$

that is an instance of an inference in ENAR that reduces C. That implies that D is false in $I(N)$, and since N is closed under ENAR the ground clause D is a reduced ground instance of N and hence a smaller counterexample, a contradiction to the minimality of C. □

A reduced ground instance $C\sigma$ of a constrained clause C is called *redundant* in N (with respect to R) if there exist reduced ground instances $C_1\sigma_1, \ldots, C_k\sigma_k$ of clauses $C_1, \ldots, C_k$ in N such that $C\sigma \succ C_i\sigma_i$ for $i = 1, \ldots, k$, and $T_R \cup \{C_1\sigma_1, \ldots, C_k\sigma_k\} \models C\sigma$.

A reduced ground instance

$$\frac{C_1\sigma \quad \ldots \quad C_n\sigma}{C\sigma} \qquad \text{of an inference} \qquad \frac{C_1 \quad \ldots \quad C_n}{C}$$

where $C_n\sigma$ is the main premise is called *redundant* in N (with respect to R) if either one of the premises $C_1\sigma, \ldots, C_n\sigma$ is redundant, or if there exist reduced ground instances $D_1\sigma_1, \ldots, D_k\sigma_k$ of N such that $C_n\sigma \succ D_i\sigma_i$ for $i = 1, \ldots, k$ and $T_R \cup \{D_1\sigma_1, \ldots, D_k\sigma_k\} \models C\sigma$. A non-ground clause or inference is redundant if all its reduced ground instances are redundant. The following well-known lemma allows to delete clauses without losing redundancy.

Lemma 7 *Let N be a set of constrained clauses and M the set of redundant clauses in N. If a constrained clause C is redundant in N then it is redundant in $N \setminus M$.*

Proof: Suppose $\hat{C}$ is redundant in N but not in $N \setminus M$. Then there exists a reduced ground instance of $\hat{C}$ that is redundant in N but not in $N \setminus M$. Let C be the least such reduced ground instance. Since C is redundant in N, there exist reduced ground instances $D_1, \ldots, D_n$ of N that together with T_R imply C. Let $D_1, \ldots, D_n$ be the least set of such reduced ground instances with respect to the multiset extension of $\succ$. Now suppose some D_i is a reduced ground instance of M but not of $N \setminus M$. Since D_i is smaller than C, it is redundant in $N \setminus M$ by induction hypothesis. That is, D_i is in turn implied by T_R and smaller reduced ground instances $D'_1, \ldots, D'_m$ of $N \setminus M$. We may replace D_i by $D'_1, \ldots, D'_m$ to obtain a smaller multiset, in contradiction to the minimality of $D_1, \ldots, D_n$. Hence $D_1, \ldots, D_n$ must be reduced ground instances of $N \setminus M$, and C is redundant in $N \setminus M$. □

A set N of clauses is called *saturated up to redundancy* (with respect to ENAR) if all inferences in ENAR from premises in N are redundant.

Lemma 8 *Let N be a set of clauses that is saturated up to redundancy with respect to* ENAR. *Then either $\bot \in N$ or $I(N) \models \mathrm{rgnd}_R(N)$.*

Proof: Suppose N is saturated up to redundancy, $\bot \notin N$, and $I(N) \not\models \mathrm{rgnd}_R(N)$. Let C be the least counterexample for $I(N)$ among the reduced ground instances of N. Since ENAR has the reduction property for counterexamples, there exists a reduced ground instance of an inference in ENAR that reduces C to some clause D that is also false in $I(N)$. Since N is saturated w.r.t. ENAR, this inference is redundant, hence D follows from T_R and reduced ground instances smaller than C. By minimality of C these are true in $I(N)$, and D must be true in $I(N)$ as well, a contradiction. □

A *theorem proving derivation* is a sequence of sets of clauses $N_0 \vdash N_1 \vdash \ldots$ such that for all steps $N_i \vdash N_{i+1}$, $i \geq 0$, either (1) $N_{i+1} = N_i \cup \{C\}$ for some constrained clause C such that from any model of $T_R \cup N_i \cup S_i \models \{C\}$ where S_i is a set of skolem axioms, or (2) $N_{i+1} = N_i \setminus \{C\}$ for some constrained clause C which is redundant in N_i, and $\bigcup_i S_i$ is fresh with respect to $T_R \cup N_0$. In case (1) we call the step a *deduction step* and in case (2) a *deletion step*. For such a derivation the set of *persistent clauses* N_∞ is defined as $N_\infty = \bigcup_{i \geq 0} \bigcap_{j \geq i} N_j$.

A simplification may be viewed as two derivation steps $\{C\} \cup N \vdash \{C, D\} \cup N \vdash \{D\} \cup N$. That is, a clause $C \in N$ may be simplified to a clause D if $T_R \cup \{C\} \cup N \models D$ and C is redundant in $\{D\} \cup N$.

For example, tautologies are always redundant, and reduction with R is a simplification. A subsumed clause that has at least one literal more than the clause that subsumes it is also redundant by this definition. However, the case where a clause subsumes one of its instances is not covered, as the individual ground instances do not decrease with respect to $\succ$. At the price of some technical complications it is possible to extend the definition to also cover that case.

We will now show that certain properties are preserved when going from N_0 to the limit N_∞ or vice-versa.

Lemma 9 *Let N be a set of constrained clauses, let I be a model of T_R, and let $N_0 \vdash N_1 \vdash \ldots$ be a theorem proving derivation. If all reduced ground instances of N_∞ are true in I then all reduced ground instances of $\bigcup_i N_i$ are true in I.*

Proof: Consider some reduced ground instance C of some clause $\hat{C}$ in $\bigcup_i N_i$. If $\hat{C}$ is in N_∞ then C is true by assumption. Otherwise $\hat{C}$ has been removed by some deletion step $N_i \vdash N_{i+1}$, hence it is redundant in N_i and thus in $\bigcup_i N_i$. Let M be the set of all redundant clause in $\bigcup_i N_i$, then by the above argument $(\bigcup_i N_i) \setminus M \subseteq N_\infty$. Hence C is redundant in N_∞ by Lemma 7, and there exist reduced ground instances of N_∞ that together with T_R imply C and are true in I. We conclude that C is true in I. □

This implies in particular that all reduced ground instances of N_0 are true in I. We say that a clause is *unconstrained* if its constraint is $\top$. A set of clauses is *unconstrained* if all its clauses are unconstrained.

Lemma 10 *Let N be a set of unconstrained clauses. Then* $\mathrm{rgnd}_R(N) \cup T_{R_t} \models \mathrm{gnd}(N)$.

Proof: Consider some ground instance C of a clause $\hat{C}$ in $\mathrm{gnd}(N)$. Then $\hat{C} = U\,[\top]$ and $C = U\sigma$. By normalizing σ with respect to R_t we obtain the ground substitution $\tau = \{x\sigma \Downarrow_{R_t} /x \mid x \neq x\sigma\}$. The instance $C' = U\tau$ is reduced and is also a ground instance of $\hat{C}$ that solves the trivial constraint $\top$, hence it is in $\mathrm{rgnd}_R(N)$. All the changes of atoms by the reduction from C to C' are covered by equivalences in T_{R_t}, hence C is a consequence of $\mathrm{rgnd}_R(N) \cup T_{R_t}$. □

Corollary 11 *Let N be a set of unconstrained clauses, and let I be a model of T_R. If I is a model of all reduced ground instances of N. Then I is a model all ground instances of N.*

Lemma 12 *Let $N_0 \vdash N_1 \vdash \ldots$ be a theorem proving derivation. If $T_R \cup N_0$ is satisfiable then $T_R \cup N_\infty$ is satisfiable.*

Proof: Suppose we are given a model I_0 of $T_R \cup N_0$. Let $S = \bigcup_i S_i$ be the set of skolem axioms used in the derivation. Then by Lemma 1 there exists a model I of $T_R \cup N_0 \cup S$. Furthermore, $T_R \cup N_0 \cup S$ is logically equivalent to $T_R \cup N_i \cup S$ for $i \geq 0$, hence $I \models N_i$ for all $i \geq 0$. We conclude $I \models N_\infty$. □

A theorem proving derivation is called *fair* (with respect to ENAR) if all inferences in ENAR from clauses in N_∞ are redundant in N_i for some $i \geq 0$. Since the conclusions of ground inferences are always smaller than the main premise, and since they imply themselves, an inference can be made redundant by a deduction step that adds its conclusion. Thus a fair derivation is obtained by considering inferences in a fair way, i.e. not delaying an inference ad infinitum, and adding their conclusion if they are not already known to be redundant by some suitable sufficient criterion.

Lemma 13 *Let $N_0 \vdash N_1 \vdash \ldots$ be a fair theorem proving derivation. Then N_∞ is saturated up to redundancy.*

Proof: By fairness we get that every inference from premises in N_∞ is redundant in $\bigcup_i N_i$, and by Lemma 7 in N_∞. □

Theorem 14 *Let $N_0 \vdash N_1 \vdash \ldots$ be a fair theorem proving derivation such that N_0 is unconstrained. Then N_0 is inconsistent if and only if N_∞ contains the empty clause.*

Proof: Since the derivation is fair, N_∞ is saturated up to redundancy. Thus either $\bot \in N_\infty$ or $I(N_\infty) \models T_R \cup \mathrm{rgnd}_R(N_\infty)$ by Lemma 8. Let S be the set of skolem axioms used in the derivation. If $\bot \in N_\infty$ then $T_R \cup N_0 \cup S$ is inconsistent, and in turn $T_R \cup N_0$ is inconsistent by Lemma 1, since S is fresh. Otherwise $I(N_\infty) \models \mathrm{rgnd}_R(N_\infty)$, $I(N_\infty) \models \mathrm{rgnd}_R(N_0)$ by Lemma 9, and $I(N_\infty) \models \mathrm{gnd}(N_0)$by Lemma 10, and $T_R \cup N_0$ is consistent. □

That is, ENAR is refutationally complete with respect to T_R.

Looking back at the proof, in particular Lemma 5, we see that we have indeed proved refutational completeness of a more restricted inference system that constrains inferences to maximal atoms :

$$\textit{Extended Resolution}\quad \frac{\neg A_1 \vee \ldots \vee \neg A_n \vee C\,[\mathcal{C}_1] \qquad B_1 \vee \ldots \vee B_m \vee D\,[\mathcal{C}_2]}{C \vee D\,[\mathcal{C}_1 \wedge \mathcal{C}_2 \wedge A_1 \approx \ldots \approx B_m \wedge A_1 \succ C \wedge A_1 \succ D]}$$

$$\textit{Extended Narrowing}\quad \frac{A \vee C\,[\mathcal{C}]}{\mathrm{cl}(A[r]_\pi \vee C)\,[\mathcal{C} \wedge U|_\pi \approx l \wedge A \succ C]}$$

where $l \Rightarrow r$ is a rule in R and $A|_\pi$ is not a variable.

The ordering constraints are interpreted by the given well-ordering $\succ$. To make this useful in practice it is of course necessary to provide a constraint solver. Note however, that it is sound to discard the ordering constraints whenever they are too hard to solve. For instance, the empty clause still indicates an inconsistency even with an unsolvable ordering constraint.

8 Example

We consider an example from set theory given by Plaisted and Zhu [12]. Suppose we have the rewrite rules

$$x \approx y \Rightarrow x \subseteq y \wedge y \subseteq x \tag{17}$$

$$x \subseteq y \Rightarrow \forall z(z \in x \rightarrow z \in y) \tag{18}$$

$$x \in y \cap z \Rightarrow x \in y \wedge x \in z \tag{19}$$

that describe a fragment of set theory.

We define the ordering on formulas as follows. We start by an ordering on atoms by letting $(s_1 \approx t_1) \succ (s_2 \subseteq t_2) \succ (s_3 \in t_3)$ for all terms $s_1, s_2, s_3, t_1, t_2, t_3$. On terms we assume some simplification ordering that is total on ground terms, for example a lexicographic path ordering. We extend it lexicographically to atoms with the same predicate symbol. Literals are ordered first with respect to their atom and then to their polarity. That is, $A \succ B$ implies $[\neg]A \succ [\neg]B$ and $\neg A \succ A$ for any atoms A and B. This is extended to clauses by the multiset extension of the literal ordering and to propositions in clause normal form by the multiset extension of the clause ordering.

This ordering is total on ground clauses, since the term ordering is total, and it is extended to atoms, literals and clauses so that this property is preserved. By the same argument it is well-founded.

We have to show that the ordering is compatible with the rewrite relation followed by instantiation. We first consider the effect of applying a rewrite rule to a single literal in a clause, and compare the instantiated clause normal forms. There are six cases, and we easily see that in each case the ordering holds :

$$\begin{aligned}
x \approx y \vee C &\succ (x \subseteq y \vee C) \wedge (y \subseteq x \vee C) \\
\neg x \approx y \vee C &\succ x \subseteq y \vee y \subseteq x \vee C \\
x \subseteq y \vee C &\succ \neg t \in x \vee t \in y \vee C \quad \text{for any ground term } t \\
\neg x \subseteq y \vee C &\succ (f(x,y) \in x \vee C) \wedge (\neg f(x,y) \in y \vee C) \\
x \in y \cap z \vee C &\succ (x \in y \vee C) \wedge (x \in z \vee C) \\
\neg x \in y \cap z \vee C &\succ \neg x \in y \vee \neg x \in z \vee C
\end{aligned}$$

The first four are covered by the precedence of the predicate sumbols, and the last two by the subterm property of the term ordering. This ordering extends to a context containing additional clauses. An atom in a proposition may lead to several occurrences of the atom in the clause normal form. Rewriting such an atom thus leads to the replacement of several atoms. We may obtain its effect on the clause normal form by chaining together several of the simple replacements. Then by transitivity the ordering is compatible with any rewrite step. Since our ordering includes the transformation to clause normal form, compatibility also holds for Extended Narrowing inferences.

Now suppose that in this theory we want to prove idempotency of intersection, i.e., $\forall x.x \cap x \approx x$. To simplify the presentation, we prove only the direction $\forall x.x \cap x \subseteq x$. We negate and skolemize and obtain $N_0 = \{\neg a \cap a \subseteq a\}$. To illustrate the model construction we also give the candidate models corresponding to the clause sets in the derivation. Since the only clause in N_0 has no positive literal, we get $P_{N_0} = \emptyset$. However, $I(N)$ is not empty, as it contains atoms such as $a \subseteq a$. We check that $C_0 = \neg(a \cap a \subseteq a)$ is false in $I(N)$ by rewriting $A_0 = a \cap a \subseteq a$:

$$\begin{aligned}
a \cap a \subseteq a &\Rightarrow (\forall x.x \in a \cap a \rightarrow x \in a) \\
&\stackrel{*}{\Rightarrow} \forall x.(x \in a \wedge x \in a) \rightarrow x \in a
\end{aligned}$$

We easily see that the normal form is a tautology, hence it is true in particular in $I(N_0)$. Then by definition A_0 is true and C_0 is false in $I(N_0)$. Thus C_0 is the least counterexample, as it is the only ground instance of a clause in N_0. This counterexample can be reduced by Extended Narrowing : By rewriting C_0 we get

$$\neg(\forall x.x \in a \cap a \rightarrow x \in a)$$

which we have to transform to clause normal form :

$$\begin{aligned}\neg(\forall x.x \in a \cap a \to x \in a) &\stackrel{*}{\Rightarrow} \exists x.(x \in a \cap a \ \wedge \neg(x \in a) \\ &\Rightarrow b \in a \cap a \ \wedge \neg b \in a\end{aligned}$$

where b is the skolem constant introduced for x. For each clause in the CNF we have an Extended Narrowing inference that has it as its conclusion :

$$\frac{\neg a \cap a \subseteq a}{b \in a \cap a} \tag{20}$$

and

$$\frac{\neg a \cap a \subseteq a}{\neg b \in a} \tag{21}$$

We pick the first inference as it is false in $I(N_0)$, and let $N_1 = N_0 \cup \{b \in a \cap a\}$. We notice that $C_1 = b \in a \cap a$ is not productive, since it is reducible to $b \in a \ \wedge \ b \in a$, hence we get another Extended Narrowing inference

$$\frac{b \in a \cap a}{b \in a}$$

which originates from both clauses of the reduct. We let $N_2 = N_1 \cup \{b \in a\}$. Now $b \in a$ is productive in $I(N_2)$, i.e., $P_{N_2} = \{b \in a\}$, and the least counterexample for $I(N_1)$ is once again C_0. To reduce it we now need the second inference above (21), and let $N_3 = N_2 \cup \{\neg b \in a\}$. The new clause becomes the least counterexample. It is irreducible and can be resolved with $b \in a$:

$$\frac{b \in a \qquad \neg b \in a}{\bot}$$

In this example we have used the model construction as our guide, choosing always the inference that reduces the least counterexample. We have done this to illustrate the model construction. In practice, however, it is usually not feasible to construct the model explicitly.

9 Conclusions and further work

We have proven the refutational completeness for Extended Narrowing and Resolution with ordering restrictions, under the proviso that a suitable ordering for the given theory exists. We have avoided the problem that skolemization does not preserve logical equivalence by adding axioms implying that equivalence. It remains to investigate how useful ENAR is in practice, with or without the ordering restrictions. To this end we are currently working on a prototype implementation of ENAR in ELAN[1] [15].

This work may be viewed as a first step towards calculi that integrate clause normal form computation with inferences. In particular for logical equivalences it seems preferable

[1] `http ://www.loria.fr/equipes/protheo/SOFTWARES/ELAN/`

to use them for rewriting, instead of destroying their structure by an initial transformation to clause normal form. ENAR is limited by the fact that the rewrite system is fixed. It will be interesting to see whether it is possible to find good inference systems where equivalences are added dynamically, to perform a kind of paramodulation on the level of propositions.

In the case where the rewrite system is positive in the sense of Dowek and Werner [7] the closure operation in our model construction resembles their construction of a premodel as a fixpoint of a functional derived from the rewrite rules. It will be interesting to further investigate this connection for the other cases in order to better understand which congruences lead to sequent calculi with the cut elimination property.

Acknowledgments

I thank Claude Kirchner for the many discussions on this paper.

Références

[1] Leo Bachmair and Harald Ganzinger. Non-clausal resolution and superposition with selection and redundancy criteria. In *Intl. Conf. on Logic Programming and Automated Reasoning*, LNCS 624, pages 273–284. Springer, 1992.

[2] Leo Bachmair and Harald Ganzinger. Associative-commutative superposition. In *Proc. 4th Int. Workshop on Conditional and Typed Rewriting*, LNCS 968, pages 1–14, Jerusalem, 1994. Springer.

[3] Leo Bachmair and Harald Ganzinger. Equational reasoning in saturation-based theorem proving. In *Automated Deduction - A Basis for Applications. Volume I*, chapter 11, pages 353–397. Kluwer, Dordrecht, The Netherlands, 1998.

[4] Leo Bachmair, Harald Ganzinger, Christopher Lynch, and Wayne Snyder. Basic paramodulation. *Information and Computation*, 121(2) :172–192, 1995.

[5] Leo Bachmair, Harald Ganzinger, and Jürgen Stuber. Combining algebra and universal algebra in first-order theorem proving : The case of commutative rings. In *Proc. 10th Workshop on Specification of Abstract Data Types*, LNCS 906, pages 1–29, Santa Margherita, Italy, 1995. Springer.

[6] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. Rapport de Recherche 3400, INRIA, 1998.

[7] Gilles Dowek and Benjamin Werner. Proof normalization modulo. Rapport de Recherche 3542, INRIA, 1998. Also in Types for proofs and programs 98, T. Altenkirch, W. Naraschewski, B. Rues (Eds.), LNCS 1657, Springer 1999, pp. 62-77.

[8] Harald Ganzinger and Uwe Waldmann. Theorem proving in cancellative abelian monoids (extended abstract). In *13th Int. Conf. on Automated Deduction*, LNAI 1104, pages 388–402, New Brunswick, NJ, USA, 1996. Springer.

[9] Jean-Marie Hullot. Canonical forms and unification. In *Proc. 5th Conf. on Automated Deduction*, LNCS 87, pages 318–334, Les Arcs, France, 1980. Springer.

[10] Robert Nieuwenhuis and Alberto Rubio. Theorem proving with ordering constrained clauses. In *Proc. 11th Int. Conf. on Automated Deduction*, LNCS 607, pages 477–491, Saratoga Springs, NY, 1992. Springer.

[11] Robert Nieuwenhuis and Alberto Rubio. Paramodulation with built-in AC-theories and symbolic constraints. *Journal of Symbolic Computation*, 23 :1–21, 1997.

[12] David A. Plaisted and Yunshan Zhu. Replacement rules with definition detection.

[13] Jürgen Stuber. Superposition theorem proving for abelian groups represented as integer modules. *Theoretical Computer Science*, 208(1–2) :149–177, 1998.

[14] Jürgen Stuber. Superposition theorem proving for commutative rings. In Wolfgang Bibel and Peter H. Schmitt, editors, *Automated Deduction - A Basis for Applications. Volume III. Applications*, chapter 2, pages 31–55. Kluwer, Dordrecht, The Netherlands, 1998.

[15] Jürgen Stuber. Experiments with an implementation of Extended Narrowing And Resolution in the rewriting language ELAN (system description). Submitted to RTA 2001, December 2000.

[16] Jürgen Stuber. *Superposition Theorem Proving for Commutative Algebraic Theories.* Dissertation, Naturwissenschaftlich-Technische Fakultät I, Universität des Saarlandes, Saarbrücken, 2000.

[17] Laurent Vigneron. Associative-commutative deduction with constraints. In *Proc. 12th Int. Conf. on Automated Deduction*, LNCS 814, pages 530–544, Nancy, France, 1994. Springer.

[18] Ulrich Wertz. First-order theorem proving modulo equations. Technical Report MPI-I-92-216, Max-Planck-Institut für Informatik, Saarbrücken, April 1992.

Table des matières

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

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