



HAL
open science

How can we prove that a proof search method is not an instance of another?

Guillaume Burel, Gilles Dowek

► **To cite this version:**

Guillaume Burel, Gilles Dowek. How can we prove that a proof search method is not an instance of another?. 2009. hal-04078083

HAL Id: hal-04078083

<https://hal.science/hal-04078083>

Preprint submitted on 22 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

How can we prove that a proof search method is not an instance of another?

Guillaume Burel¹ and Gilles Dowek²

¹ Nancy-Université and LORIA

guillaume.burel@ens-lyon.org, <http://www.loria.fr/~burel>

² École polytechnique and INRIA

LIX, École polytechnique, 91128 Palaiseau Cedex, France.

gilles.dowek@polytechnique.edu, <http://www.lix.polytechnique.fr/~dowek>

Abstract. We introduce a method to prove that a proof search method is not an instance of another. As an example of application, we show that Polarized resolution modulo, a method that mixes clause selection restrictions and literal selection restrictions, is not an instance of Ordered resolution with selection.

1 Introduction

An important property of the resolution method [4] is its refutational completeness, that is the possibility to derive the empty clause from any unsatisfiable set of clauses. However, the search space to derive this clause can be unnecessarily big. For instance, to derive the empty clause from the set formed with the clause P , the clause $\neg P, Q$ and the clause $\neg Q$, we can first generate the clause Q from P and $\neg P, Q$ and then the empty clause from this clause and $\neg Q$. Alternatively, we can generate the clause $\neg P$ from $\neg P, Q$ and $\neg Q$ and then the empty clause from this clause and P . These two derivations are redundant and eliminating such redundancies is a key issue to design an efficient proof search method. Several types of restrictions may be applied to eliminate these redundancies, while preserving refutational completeness.

First, we may impose some restriction on the choice of clauses, allowing the resolution rule to be applied to some pairs of clauses and forbidding it to be applied to others. This type of restriction is used, for instance, in the *Hyper-resolution* method [5], in the *Set-of-support* method [7], and in the *Semantic resolution method* [6]. For instance, in the Set-of-support method, we identify a consistent subset of the set of clauses to be refuted, called *the theory*, for instance, the subset formed with the clauses P and $\neg P, Q$. Then, the resolution rule can be applied to a pair of clauses if at most one clause is in this set, but it is forbidden if both are.

Then, we may impose some restriction on the choice of literals, allowing the resolution rule to be applied to some literals of the resolved clauses and forbidding it to be applied to others. This type of restriction is used, for instance, in *Ordered resolution with selection*. In this method, a selection function and an

order relation define a set of selected literals in each clause. Then, the resolution rule may be applied to a pair of clauses if the resolved literal are selected in the clauses, but it is forbidden otherwise.

It is easy to remark that combining clause selection restrictions and literal selection restrictions in the theory clauses may jeopardize completeness, even when the theory is consistent and the selected literals are defined with respect to an order, as in the Ordered resolution with selection.

Example 1. Consider the clauses

$$\begin{array}{l} \text{Theory} \left\{ \begin{array}{l} \underline{P} \vee Q \\ \neg \underline{P} \vee Q \end{array} \right. \\ \text{Other clauses} \{ \neg Q \end{array}$$

where the selected literals are underlined. We cannot derive the empty clause if we restrict the application of the resolution rule to clauses such that at most one of them is the theory and the resolved literal in a theory clause is selected. However, the theory is consistent, so the Set-of-support restriction alone is complete; and literals are selected according to the order $Q \prec P$, so the Ordered resolution with selection alone also is complete.

2 Polarized resolution modulo

To prove the completeness of the combination of the Set-of-support method and the Ordered resolution with selection method, we therefore need a stronger condition than the consistency of the theory and the use of an order to define selected literals. As we shall see, this condition is exactly the cut elimination property for the polarized sequent calculus modulo some rewrite rules corresponding to the theory clauses.

Indeed, the recently introduced *Polarized resolution modulo* method [3], combines these two restrictions. In this method, we first identify a subset of the set of clauses to be refuted. This set is called *the theory* and its elements *one-way clauses*. Then, in each of these clause, we identify a *selected literal*, and we impose the following restrictions:

- the resolution rule may be applied to a pair of clauses if at most one clause is a one-way clause, but it is forbidden when both are,
- the resolution rule can be applied to a pair of clauses containing a one-way clause, if the resolved literal in the one-way clause is the selected one, but not otherwise.

A last feature of Polarized resolution modulo is that unification is replaced by equational unification, but we shall not use this here.

Example 2. Consider an arbitrary set of clauses containing the clauses

$$\underline{P}, Q$$

$$\underline{P}, \neg Q$$

then taking all the clauses of this subset to be one-way clauses and selecting the underlined literals is a complete restriction of resolution.

Example 3. Consider an arbitrary set of clauses containing the clauses

$$\begin{array}{c} \underline{\neg \varepsilon(x \dot{\vee} y)}, \varepsilon(x), \varepsilon(y) \\ \varepsilon(x \dot{\vee} y), \neg \varepsilon(x) \\ \varepsilon(x \dot{\vee} y), \neg \varepsilon(y) \\ \underline{\neg \varepsilon(\dot{\neg} x)}, \neg \varepsilon(x) \\ \varepsilon(\dot{\neg} x), \varepsilon(x) \\ \underline{\neg \varepsilon(\dot{\forall}_T x)}, \varepsilon(x y) \\ \varepsilon(\dot{\forall}_T x), \neg \varepsilon(x H_T(x)) \\ \underline{\neg \varepsilon(Null(S x))} \\ \varepsilon(Null 0) \end{array}$$

and clauses containing no occurrences of the symbols H_T . Then, taking all these clauses of this subset to be one-way clauses and selecting the underlined literals is a complete restriction of resolution.

Replacing unification with equational unification makes this method complete for Simple Type Theory [3].

As we saw in Example 1, this method is not always complete even if the theory is consistent and if the selected literals are maximal for some order on atoms. But, we have proved in [3] that the completeness of this method is equivalent to cut elimination for the polarized sequent calculus modulo the rewrite system associated to the set of one-way clauses, where the rules of the polarized sequent calculus modulo are given in Fig. 1 and the relation between clauses with selected literals and polarized rewrite rules is defined as follows.

Definition 1. Let T be a set of clauses, such that in each clauses, a literal is selected. The rewrite system associated with T is defined by:

To each selected literal L in a clause L, C_1, \dots, C_p corresponds a rewrite rule

- if L is a negative literal $\neg P$, the rule $P \longrightarrow_- \forall x_1 \dots \forall x_n (C_1 \vee \dots \vee C_p)$
- if L is a positive literal P , the rule $P \longrightarrow_+ \neg \forall x_1 \dots \forall x_n (C_1 \vee \dots \vee C_p)$

where x_1, \dots, x_n are the variables free in C but not in P .

Theorem 1. Let T be a set of one-way clauses and \mathcal{R} be the rewrite system associated with T . Polarized resolution modulo with the set of one-way clauses T is complete if and only if the polarized sequent calculus modulo \mathcal{R} admits the cut rule.

$$\begin{array}{l}
\overline{A \vdash B} \text{ axiom if } A \rightarrow_{-}^{*} P, B \rightarrow_{+}^{*} P \text{ and } P \text{ atomic} \\
\frac{\Gamma, B \vdash \Delta \quad \Gamma \vdash C, \Delta}{\Gamma \vdash \Delta} \text{ cut if } A \rightarrow_{-}^{*} B, A \rightarrow_{+}^{*} C \\
\frac{\Gamma, B, C \vdash \Delta}{\Gamma, A \vdash \Delta} \text{ contr-left if } A \rightarrow_{-}^{*} B, A \rightarrow_{-}^{*} C \\
\frac{\Gamma \vdash B, C, \Delta}{\Gamma \vdash A, \Delta} \text{ contr-right if } A \rightarrow_{+}^{*} B, A \rightarrow_{+}^{*} C \\
\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \text{ weak-left} \\
\frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta} \text{ weak-right} \\
\frac{}{\Gamma, A \vdash \Delta} \perp\text{-left if } A \rightarrow_{-}^{*} \perp \\
\frac{\Gamma \vdash B, \Delta}{\Gamma, A \vdash \Delta} \neg\text{-left if } A \rightarrow_{-}^{*} \neg B \\
\frac{\Gamma, B \vdash \Delta}{\Gamma \vdash A, \Delta} \neg\text{-right if } A \rightarrow_{+}^{*} \neg B \\
\frac{\Gamma, B \vdash \Delta \quad \Gamma, C \vdash \Delta}{\Gamma, A \vdash \Delta} \vee\text{-left if } A \rightarrow_{-}^{*} (B \vee C) \\
\frac{\Gamma \vdash B, C, \Delta}{\Gamma \vdash A, \Delta} \vee\text{-right if } A \rightarrow_{+}^{*} (B \vee C) \\
\frac{\Gamma, C \vdash \Delta}{\Gamma, A \vdash \Delta} \langle x, B, t \rangle \forall\text{-left if } A \rightarrow_{-}^{*} \forall x B, (t/x)B \rightarrow_{-}^{*} C \\
\frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A, \Delta} \langle x, B \rangle \forall\text{-right if } A \rightarrow_{+}^{*} \forall x B, x \notin FV(\Gamma \Delta)
\end{array}$$

Fig. 1. Polarized sequent calculus modulo

Proof. Corollary of Theorem 1 of [3].

For instance, the polarized sequent calculus modulo the rewrite system

$$\begin{array}{l} P \longrightarrow_+ \neg Q \\ P \longrightarrow_+ \neg\neg Q \end{array}$$

has the cut elimination property (Proposition 7 of [2] proves that this property holds whenever the left hand sides of the positive and of the negative rules are disjoint sets) hence the completeness in Example 2.

In the same way, the polarized sequent calculus modulo the rewrite system

$$\begin{array}{ll} \varepsilon(x \dot{\vee} y) \longrightarrow_- (\varepsilon(x) \vee \varepsilon(y)) & \varepsilon(x \dot{\vee} y) \longrightarrow_+ \neg\neg\varepsilon(x) \\ \varepsilon(x \dot{\vee} y) \longrightarrow_+ \neg\neg\varepsilon(y) & \\ \varepsilon(\dot{\wedge} x) \longrightarrow_- \neg\varepsilon(x) & \varepsilon(\dot{\wedge} x) \longrightarrow_+ \neg\varepsilon(x) \\ \varepsilon(\dot{\forall}_T x) \longrightarrow_- \forall y \varepsilon(x y) & \varepsilon(\dot{\forall}_T x) \longrightarrow_+ \neg\neg\varepsilon(x H_T(x)) \\ \varepsilon(\text{Null } 0) \longrightarrow_+ \neg\perp & \\ \varepsilon(\text{Null } (S x)) \longrightarrow_- \perp & \end{array}$$

has the cut elimination property for all sequents containing no occurrences of the symbols H_T , hence the completeness in Example 3.

3 Comparing Polarized resolution modulo with Set-of-support resolution

In contrast, the polarized sequent calculus modulo the rewrite system

$$\begin{array}{l} P \longrightarrow_- Q \\ P \longrightarrow_+ \neg Q \end{array}$$

does not have the cut elimination property. Indeed, the proposition Q has a proof by cutting through P , but no cut free proof. This explains the incompleteness in Example 1.

The next example shows that Polarized resolution modulo can fail in finite time for some input when Set-of-support resolution loops.

Example 4. Consider the one-way clause

$$\underline{P(f(x))}, \neg P(x)$$

and another clause

$$P(a)$$

for some constant a . Both Polarized resolution modulo and Set-of-support resolution are complete. However, Polarized resolution modulo fails in finite time to derive the empty clause (the two clauses cannot be resolved due to the condition on the selected literal), whereas Set-of-support resolution loops, deriving the clauses $P(\underbrace{f(\dots(f(a)\dots)}_{i \text{ times}}))$ for all $i > 0$.

In this example, we see the importance of having a selection function. However, having such a selection function is not enough, as we will see in Section 4.

4 Comparing Polarized resolution modulo with Ordered resolution with selection

Ordered resolution with selection [1] is a proof search method parametrized by a computable selection function S , that associates to each clause a set of negative literals of this clause and a decidable order relation \prec on atoms that is stable by substitution and total on ground atoms. Each pair $\langle S, \prec \rangle$ defines a different proof-search method. Thus Ordered resolution with selection is a family of proof search methods rather than a single method.

Many known restrictions of resolution are instances of Ordered resolution with selection for an appropriate selection function and order. Thus, we may wonder if Polarized resolution modulo is, in the same way an instance of Ordered resolution with selection. We prove now that this is not the case.

We prove more generally, that if m and m' are two proof-search methods and T is a theory such that the completeness of m can be proved in T and m' fails in finite time attempting to prove a contradiction in T , then m and m' are different methods: they are separated by the theory T .

4.1 Separation of proof-search methods

Consider a decidable set of axioms T , that is an ω -consistent extension of arithmetic. We can express in the language of T , a proposition Bew with two free variables, such that if U is a decidable set of axioms (i.e. the index of a total computable function characterizing these axioms) and A is a proposition (i.e. the index of a proposition) then the sequent $T \vdash Bew(U, A)$ is provable in predicate logic if and only if the sequent $U \vdash A$ is.

Consider a proof-search method m . We can build, in the language of T , a proposition M with two free variables, such that if U is a decidable set of axioms and A is a proposition then

- if the method m applied to the theory U and to the proposition A succeeds then the sequent $T \vdash M(U, A)$ is provable,
- if the method m applied to the theory U and to the proposition A does not succeed (i.e. fails in finite time or loops) then the sequent $T \vdash M(U, A)$ is not provable,
- if the method m applied to the theory U and to the proposition A fails in finite time then the sequent $T \vdash \neg M(U, A)$ is provable.

Assume, moreover that the completeness of the method m is provable in T , i.e. that the sequent

$$T \vdash \forall U \forall A (Bew(U, A) \Rightarrow M(U, A))$$

is provable in predicate logic. Then, by Gödel second incompleteness theorem, the sequent

$$T \vdash \neg Bew(T, \perp)$$

is not provable in predicate logic, hence the sequent

$$T \vdash \neg M(T, \perp)$$

is not provable either. Thus, if the completeness of m can be proved in the theory T , then the method m attempting to prove \perp in the theory T cannot fail in finite time: it must loop.

If m' is a proof-search method that fails in finite time attempting to prove \perp in the theory T , then m and m' are different.

4.2 Translations

This can be generalized in the following way. Assume that ϕ is a translation from theories to theories whose completeness can be proved in T , i.e. such that the sequent

$$T \vdash \forall U (Bew(U, \perp) \Rightarrow Bew(\phi(U), \perp))$$

is provable. Assume, moreover that the completeness of the method m , is provable in T , i.e. that the sequent

$$T \vdash \forall U \forall A (Bew(U, A) \Rightarrow M(U, A))$$

is provable in predicate logic. Then, by Gödel second incompleteness theorem, the sequent

$$T \vdash \neg Bew(T, \perp)$$

is not provable in predicate logic, hence the sequent

$$T \vdash \neg M(\phi(T), \perp)$$

is not provable either. Thus, if the completeness of the translation ϕ and that of the method m can be proved in T , then the method m attempting to prove \perp in the theory $\phi(T)$ cannot fail it finite time: it must loop.

If m' is a proof-search method that fails in finite time attempting to prove \perp from T , then the application of m' to a theory U is not the application of m to $\phi(U)$.

4.3 Application to Ordered resolution with selection

Let \mathcal{H} be the first-order presentation of Simple type theory of Example 3. The completeness of Ordered resolution with selection can be proved in \mathcal{H} provided the stability and the totality of the order are provable in \mathcal{H} .

Attempting to prove \perp in the theory \mathcal{H} in Polarized resolution modulo fails in finite time. Thus, Polarized resolution modulo is not an instance of Ordered resolution with selection for any selection function and order relation whose stability and totality can be proved in \mathcal{H} . Neither it is the application of Ordered resolution with selection to a translation of its input, for a translation provably complete in \mathcal{H} .

5 Conclusion

Polarized resolution modulo is a combination of a restriction on the choice of clauses and a restriction on the choice of literals in resolution. Combining these two restrictions makes the method so restrictive that it is not always complete, even when the one-way clauses form a consistent subset and the selection of literals is based on an ordering on atoms. But the completeness condition is stronger, as completeness is equivalent to the cut elimination property for the associated sequent calculus in Polarized deduction modulo. Thus, unlike Ordered resolution with selection, the completeness of all instances of the method cannot be proved in the same theory.

The advantage of such methods whose completeness of all instances cannot be proved in the same theory is that their logical strength is not limited by the logical strength of the theory in which the completeness of the method can be proven.

References

1. L. Bachmair and H. Ganzinger. Resolution Theorem Proving, J.A. Robinson and A. Voronkov, *Handbook of Automated Reasoning*, Elsevier, 2001.
2. G. Dowek. What is a theory? H. Alt and A. Ferreira, *Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 2285, Springer, 2002, pp. 50-64.
3. G. Dowek. Polarized resolution modulo. Manuscript, 2009.
4. J.A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12, 1965, pp. 23–41.
5. J.A. Robinson. Automatic deduction with hyper-resolution. *International Journal of Computer Mathematics*, 1, 1965, pp. 227–234.
6. J.R. Slagle. Automatic theorem proving with renamable and semantic resolution. *J. ACM*, 14, 1967, pp. 687–697.
7. L. Wos, G.A. Robinson, D.F. Carson. Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM*, 12, 1965, pp. 536–541.