

Cut Elimination in Deduction Modulo by Abstract Completion

Guillaume Burel, Claude Kirchner

▶ **To cite this version:**

Guillaume Burel, Claude Kirchner. Cut Elimination in Deduction Modulo by Abstract Completion. Symposium on Logical Foundations of Computer Science LFCS'07, Sergei Artemov, Jun 2007, New York, United States. pp.115-131, 10.1007/978-3-540-72734-7_9 . inria-00115556v3

HAL Id: inria-00115556

https://inria.hal.science/inria-00115556v3

Submitted on 26 Feb 2007

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.

inria-00115556, version 3 - 26 Feb 2007

Cut Elimination in Deduction Modulo by Abstract Completion

Guillaume Burel[1] and Claude Kirchner[2]

[1] Université Henri Poincaré & LORIA[3]
guillaume.burel@ens-lyon.org
[2] INRIA & LORIA[3]
Claude.Kirchner@loria.fr
[3] UMR 7503 CNRS-INPL-INRIA-Nancy2-UHP

Abstract. Deduction Modulo implements Poincaré's principle by identifying deduction and computation as different paradigms and making their interaction possible. This leads to logical systems like the sequent calculus or natural deduction modulo. Even if deduction modulo is logically equivalent to first-order logic, proofs in such systems are quite different and dramatically simpler with one cost: cut elimination may not hold anymore. We prove first that it is even undecidable to know, given a congruence over propositions, if cuts can be eliminated in the sequent calculus modulo this congruence.
Second, to recover the cut admissibility, we show how computation rules can be added following the classical idea of completion *a la* Knuth and Bendix. Because in deduction modulo, rewriting acts on terms as well as on propositions, the objects are much more elaborated than for standard completion. Under appropriate hypothesis, we prove that the sequent calculus modulo is an instance of the powerful framework of *abstract canonical systems* and that therefore, cuts correspond to critical proofs that abstract completion allows us to eliminate.
In addition to an original and deep understanding of the interactions between deduction and computation and of the expressivity of abstract canonical systems, this provides a mechanical way to transform a sequent calculus modulo into an equivalent one admitting the cut rule, therefore extending in a significant way the applicability of mechanized proof search in deduction modulo.

Key words: Knuth-Bendix completion, automated deduction and interactive theorem proving, cut elimination, deduction modulo, proof ordering, abstract canonical system

1 Introduction

The complementarity and interaction between computation and deduction is known since at least Henri Poincaré, and deduction modulo [16] is a way to present first-order logic taking advantage from this complementarity. Deduction

modulo is at the heart of proof assistants and proof search methods, either implicitly or explicitly (see for instance [24,3,16,5]) and getting a deep understanding of its logical behavior is of prime interest either for theoretical or practical purposes.

In deduction modulo, computations are modeled by a congruence relation between terms and between propositions. The logical deductions are done modulo this congruence that is often better represented by a rewrite relation over first-order terms and propositions, leading to the asymmetric sequent calculus [14].

In the sequent calculus modulo, the Hauptsatz, *i.e.* the fact that cuts are not needed to build proofs, is no longer true as one can see from an example derived from Crabbé's proof of the non-normalization of Zermelo's theory [8] (see for instance [16]). But we know that the admissibility of the cut rule is fundamental for at least two related reasons: first, if a system admits the cut rule, then the formulæ needed to build a sequent calculus proof of some sequent are subformulæ[4] of the ones appearing in it, so that the search space is, in a sense, limited. Such proofs are sometimes called *analytic* [14]. The tableaux method is based on this fact, and for instance TaMeD [5], a tableaux method based on deduction modulo, is shown to be complete only for cut-free systems. On the other hand, it has been shown [21] that a proof search method for deduction modulo like ENAR [16]—which generalizes resolution and narrowing—is equivalent to the cut-free fragment of deduction modulo. ENAR is therefore complete if and only if the cut rule is admissible.

So on the one hand, we like to have a powerful congruence but this may be at the price of loosing cut admissibility. How can we get both? Gilles Dowek has shown [14] that cut admissibility is equivalent to the confluence of the rewrite system, provided only first-order *terms* are rewritten. It is no longer true when *propositions* are also rewritten, and the cut admissibility is in that case a stronger notion than confluence. Therefore he wanted to build a generalized completion procedure whose input is a rewrite system over first-order terms and propositions and that computes a rewrite system such that the associated sequent calculus modulo admits the cut rule. Such a completion procedure was proposed for the quantifier free case in [13], based on the construction of a model for the theory associated with the rewrite system.

To fully solve this question, including *unlimited* use of quantifiers, we propose here a quite different approach based on the notion of abstract canonical system and inference introduced in [12,4]. This abstract framework is based on a proof ordering whose goal is to apprehend the notion of proof quality from which the notions of canonicity, completeness and redundancy follow up. It is shown to be adapted to existing completion procedures such as ground completion [10] and standard (a.k.a. Knuth-Bendix [23]) completion [6].

[4] In the case of deduction modulo, the intuitive notion of subformula must of course take into account the equivalence relation.

To present the general idea of our approach, let us consider the simple example of Crabbé's axiom [8] $A \Leftrightarrow B \wedge \neg A$.[5] Can we find, for the sequent calculus modulo the rewrite system $A \to B \wedge \neg A$, a provable sequent without any cut-free proof? Indeed, let us try to build a minimal example. We will show in Prop. 4 that such a proof, in its simplest form, is necessarily of the shape:

$$\dfrac{\dfrac{\vdots}{\dfrac{A, B \wedge \neg A \vdash}{A \vdash}\uparrow\text{-l}} \quad \dfrac{\vdots}{\dfrac{\vdash B \wedge \neg A, A}{\vdash A}\uparrow\text{-r}}}{\vdash}\,\mathsf{Cut}(A)$$

where the rules labeled "↑-r" and "↑-l" allow to apply the oriented axioms respectively on the right or on the left. In order to validate this proof pattern, we have to check if it is possible to close both sides of the proof tree, possibly adding informations in the initial sequent.

First, we can trivially close the left part as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A, B \vdash A}\,\mathsf{Axiom}}{A, B, \neg A \vdash}\,\neg\text{-l}}{A, B \wedge \neg A \vdash}\,\wedge\text{-l}} \quad .$$

Second, to close the right part, we must have a proof in the form:

$$\dfrac{\vdash B, A \qquad \dfrac{\dfrac{}{A \vdash A}\,\mathsf{Axiom}}{\vdash \neg A, A}\,\neg\text{-r}}{\vdash B \wedge \neg A, A}\,\wedge\text{-r} \quad .$$

To enforce the proof of $\vdash B, A$, we must add either A or B to the left of the sequent, and we only have to consider B, since we have cut around A. We obtain the critical proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A, B \vdash A}\,\mathsf{Axiom}}{A, B, \neg A \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A \vdash}\,\wedge\text{-l}}{B, A \vdash}\,\uparrow\text{-l} \qquad \dfrac{\dfrac{\dfrac{}{B \vdash B, A}\,\mathsf{Axiom} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\mathsf{Axiom}}{B \vdash \neg A, A}\,\neg\text{-r}}{B \vdash B \wedge \neg A, A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r}}{B \vdash}\,\mathsf{Cut}(A) \quad .$$

We can also easily show that there is no cut-free proof of $B \vdash$, simply because no inference rule is applicable to it except Cut. If we want to have a cut-free proof, we need to make B reducible by the congruence, hence the idea to complete the initial system with a new rule which is a logical consequence of the current system. In our case, we must therefore add the rule $B \to \bot$.

[5] In [8], A represents $r_s \in r_s$ and B is $r_s \in s$ where $r_s \stackrel{!}{=} \{x \in s : x \notin x\}$. Then, there is a proof of $r_s \notin s$ in Zermelo's set theory that is not normalizing.

With this new rule, we will show that there is no more critical proof and that therefore the sequent calculus modulo the proposition rewrite system

$$\begin{cases} A \to B \wedge \neg A \\ B \to \bot \end{cases}$$

admits the cut rule and has the same expressive power as the initial one.

The study of this question indeed reveals general properties of the sequent calculus modulo and our contributions are the following:

- We provide an appropriate Noetherian ordering on the proofs of the sequent calculus modulo a rewrite system; This ordering allows us to set on the proof space of sequent calculus modulo a structure of abstract canonical system;
- We characterize the critical proofs in deduction modulo as simple cuts;
- By an appropriate correspondence between sequents and rewrite systems, we establish a precise correspondence between the limit of a completion process and a cut free sequent calculus;
- We show the applicability of the general results, in particular on sequent calculus modulo rewrite systems involving quantifiers, therefore generalizing all previously known results;
- We establish the limits of our approach by proving the undecidability of cut admissibility and of the search for critical proofs.

As an important by-product of these results, we demonstrate the expressive power of abstract canonical systems (ACS for short).

The next section will present the minimal knowledge needed on deduction modulo and abstract canonical systems to make the paper self-contained, and states the undecidability of the admissibility of the cut rule in deduction modulo. In Sect. 3, we show how to set, on the proof space of sequent calculus modulo, a structure of abstract canonical system. In particular we make precise why the postulates of ACS are fulfilled. This allows us in Sect. 4 to characterize the critical proofs of deduction modulo and to set-up the completion process as the appropriate (and indeed non-trivial) instance of the abstract completion process. We also provide an algorithm to systematically transform a set of sequents into an appropriate set of proposition rewrite rules, therefore making the whole framework operational. We conclude after presenting in more details Crabbé's example as well as several examples involving quantifiers. All proofs can be found in the full version of this paper [7].

2 Prerequisites

2.1 Rewritings

We define here how propositions are rewritten in deduction modulo.

We use standard definitions for terms, predicates, propositions (with connectors $\neg, \Rightarrow, \wedge, \vee$ and quantifiers $\forall, \exists$), substitutions, term rewrite rules and term rewriting, as can be found in [2,19]. The set of terms built from a signature Σ

and a set of variables V is denoted by $\mathcal{T}(\Sigma, V)$, the replacement of a variable x by a term t in a proposition P by $\{t/x\}P$, the application of a substitution σ in a proposition P by σP.

An atomic proposition $A(s_1, \ldots, s_i, \ldots, s_n)$ can be rewritten to the atomic proposition $A(s_1, \ldots, t_i, \ldots, s_n)$ by a term rewrite rule $l \to r$ if s_i can be rewritten to t_i by $l \to r$.

A *proposition rewrite rule* is the pair of an atomic proposition A and a proposition P, such that all free variables of P appear in A. It is denoted $A \to P$. A *proposition rewrite system* is a set of proposition rewrite rules. The set of all proposition rewrite systems is denoted $\mathcal{PRS}$.

An atomic proposition A can be rewritten to a proposition P by a proposition rewrite rule $B \to Q$ if there exists some substitution σ such that $\sigma B = A$ and $\sigma Q = P$. Semantically, this proposition rewrite relation must be seen as a logical equivalence between propositions.

Note that we do not define how to rewrite non-atomic propositions by proposition rewrite rules, as in [16], because this can be simulated in the sequent calculus modulo we present in the next section.

In the following, the term rewrite system used in addition to all the proposition rewrite systems we will consider is fixed. It is supposed to be *terminating and confluent*. It will be denoted $R_{\mathcal{T}(\Sigma,V)}$.

The *subformula relation* $\succ$ is the least transitive relation such that:

- $P \succ P_i$ if $P = P_1 \wedge P_2$, $P = P_1 \vee P_2$ or $P = \neg P_1$;
- $P \succ \{t/x\}Q$ if $P = \forall x.\ Q$ or $P = \exists x.\ Q$;
- $P \succ Q$ if P can be rewritten to Q by $R_{\mathcal{T}(\Sigma,V)}$

for all terms t, variables x and propositions P, Q, P_1, P_2. It is well-founded because of the termination of $R_{\mathcal{T}(\Sigma,V)}$.

2.2 Sequent Calculus Modulo

Sequent calculus modulo can be seen as an extension of the sequent calculus of Gentzen [20]. We will use the denominations of [19].

A *sequent* is a pair of multisets of propositions Γ, Δ. It is denoted by $\Gamma \vdash \Delta$. The sets of all sequents will be denoted $\mathcal{S}$. For a sequent $\Gamma \vdash \Delta$, if $x_1, \ldots, x_n$ are the free variables of Γ, Δ, we will denote by $\mathcal{P}(\Gamma \vdash \Delta)$ the proposition $\forall x_1, \ldots, x_n.\ (\bigwedge \Gamma \Rightarrow \bigvee \Delta)$.

In Fig. 1 we present some inference rules of our *sequent calculus modulo*. They differ from the ones of [14] because the congruence is externalized through specific inference rules $\uparrow$-l and $\uparrow$-r (as can be found in [21]), but there is no contraction or weakening rules. The other logical rules are the one of the standard sequent calculus. For $\forall$-l and $\exists$-r, the quantified formula that is decomposed is kept. *Proofs* are trees labeled by sequents built using these rules, and where all leaves are Axioms. The root sequent is called the *conclusion*. A proof is said to be built in the proposition rewrite system R if all $\uparrow$-l and $\uparrow$-r use only rules that appear in $R \cup R_{\mathcal{T}(\Sigma,V)}$. The set of all proofs will be denoted by $\mathcal{SQM}$.

Identity Group: $\dfrac{}{\Gamma, P \vdash P, \Delta}\ \mathsf{Axiom}$ $\qquad$ $\dfrac{\Gamma, P \vdash \Delta \quad \Gamma \vdash P, \Delta}{\Gamma \vdash \Delta}\ \mathsf{Cut}(P)$

Logical Rules: $\dfrac{\Gamma \vdash P, \Delta}{\Gamma, \neg P \vdash \Delta}\ \neg\text{-l}$ $\qquad$ $\dfrac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \Rightarrow Q \vdash \Delta}\ \Rightarrow\text{-l}$

$\dfrac{\Gamma \vdash \{c/x\}P, \Delta}{\Gamma \vdash \forall x.\ P, \Delta}\ \forall\text{-r}$ c free in Γ, Δ $\qquad$ $\dfrac{\Gamma \vdash \exists x.\ P, \{t/x\}P, \Delta}{\Gamma \vdash \exists x.\ P, \Delta}\ \exists\text{-r}$ $t \in \mathcal{T}(\Sigma, V)$

Rewrite Rules: if A can be rewritten to P, either by a term or a proposition rewrite rule (in one step),

$$\frac{\Gamma, A, P \vdash \Delta}{\Gamma, A \vdash \Delta}\ \uparrow\text{-l} \qquad \frac{\Gamma \vdash A, P, \Delta}{\Gamma \vdash A, \Delta}\ \uparrow\text{-r}$$

Fig. 1. Some inference rules of the sequent calculus modulo.

$\mathsf{Cut}(P)$ permits essentially to extend the proof search space with the proposition P. Logical Rules decompose some proposition which is called *principal*. Rewrite Rules, that do not appear in Gentzen's sequent calculus, introduce proposition rewriting into the proof system. Note that only atomic propositions are rewritten, and that we keep the original formula in the sequent.

A proposition rewrite system R is said to *admit* Cut if for all sequents $s \in \mathcal{S}$, s has a proof in R if and only if s has a proof in R without using Cut. It is well-known (Gentzen's Hauptsatz [20], or more accurately [14] because of $R_{\mathcal{T}(\Sigma,V)}$) that $\emptyset$ admits Cut.

It is important to be aware that free variables appearing in a sequent play the same role as fresh constants, because no inference rules can modify them. As a consequence, one can restrict oneself to closed sequents, as indicated in [16, Proposition 1.5].

Proposition 1 (Equivalence). *The sequent calculus modulo (partly) presented in Fig. 1 is equivalent to the Asymmetric Sequent Calculus Modulo of [14].*

In particular, our system has the weakening and the contraction properties:

- if there exists a proof of $\Gamma \vdash \Delta$, then for all propositions P there exist proofs of $\Gamma, P \vdash \Delta$ and $\Gamma \vdash P, \Delta$;
- there exists a proof of $\Gamma, P \vdash \Delta$ if and only if there exists a proof of $\Gamma, P, P \vdash \Delta$, and there exists a proof of $\Gamma \vdash P, \Delta$ if and only if there exists a proof of $\Gamma \vdash P, P, \Delta$.

Our sequent calculus also satisfies Kleene's Lemma:

Lemma 1 (Kleene Lemma [21, Lemme 3.3]). *If a sequent, containing the non-atomic formula P, has a proof (resp. cut-free proof) in R, then it has a proof (resp. cut-free proof) in R whose first rule is a logical rule with principal proposition P.*

We prove also the following new result:

Theorem 1 (Undecidability of the Cut Admissibility). *Given a propositional rewrite system $\mathcal{R}$, it is undecidable to know if $\mathcal{R}$ admits* Cut.

2.3 Abstract Canonical Systems and Inference

The results in this section are extracted from [11,12,4], which should be consulted for motivations, details and proofs.

Let $\mathbb{A}$ be the set of all formulæ over some fixed vocabulary. Let $\mathbb{P}$ be the set of all proofs. These sets are linked by two functions: $[\cdot]^{Pm} : \mathbb{P} \to 2^{\mathbb{A}}$ gives the *premises* in a proof, and $[\cdot]_{Cl} : \mathbb{P} \to \mathbb{A}$ gives its *conclusion*. Both are extended to sets of proofs in the usual fashion. The set of proofs built using assumptions in $A \subseteq \mathbb{A}$ is denoted by

$$Pf(A) \quad \stackrel{!}{=} \quad \left\{p \in \mathbb{P} : [p]^{Pm} \subseteq A\right\} \ .$$

The framework described here is predicated on two *well-founded* partial orderings over $\mathbb{P}$: a *proof ordering* $>$ and a *subproof relation* $\rhd$. They are related by a monotonicity requirement (postulate E). We assume for convenience that the proof ordering only compares proofs with the same conclusion ($p > q \Rightarrow [p]_{Cl} = [q]_{Cl}$), rather than mention this condition each time we have cause to compare proofs.

We will use the term *presentation* to mean a set of formulæ, and *justification* to mean a set of proofs. We reserve the term *theory* for deductively closed presentations:

$$Th\, A \quad \stackrel{!}{=} \quad [Pf(A)]_{Cl} \quad = \quad \{[p]_{Cl} : p \in \mathbb{P},\ [p]^{Pm} \subseteq A\} \ .$$

Presentations A and B are *equivalent* ($A \equiv B$) if their theories are identical: $Th\, A = Th\, B$. In addition to this, we assume the two following postulates:

Postulate A (Reflexivity) *For all presentations A:*

$$A \subseteq Th\, A$$

Postulate B (Closure) *For all presentations A:*

$$Th\ Th\, A \subseteq Th\, A$$

We call a proof *trivial* when it proves only its unique assumption and has no subproofs other than itself, that is, if $[p]^{Pm} = \{[p]_{Cl}\}$ and $p \trianglerighteq q \Rightarrow p = q$, where $\trianglerighteq$ is the reflexive closure of the subproof ordering $\rhd$. We denote by $\widehat{a}$ such a trivial proof of $a \in \mathbb{A}$ and by $\widehat{A}$ the set of trivial proofs of each $a \in A$.

We assume that proofs use their assumptions (postulate C), that subproofs don't use non-existent assumptions (postulate D), and that proof orderings are monotonic with respect to subproofs (postulate E):

Postulate C (Trivia) *For all proofs p and formulæ a:*

$$a \in [p]^{Pm} \Rightarrow p \trianglerighteq \widehat{a}$$

Postulate D (Subproofs Premises Monotonicity) *For all proofs p and q:*

$$p \trianglerighteq q \Rightarrow [p]^{Pm} \supseteq [q]^{Pm}$$

Postulate E (Replacement) *For all proofs p, q and r:*

$$p \rhd q > r \Rightarrow \exists v \in Pf([p]^{Pm} \cup [r]^{Pm}).\ p > v \rhd r$$

Postulate E essentially says that replacing one of its subproof by a smaller proof makes a proof smaller. However, the proof v is not necessarily obtained by syntactically replacing q by r in p.

We make no other assumptions regarding proofs or their structure and the proof ordering $>$ is lifted to a quasi-ordering $\gtrsim$ over presentations:

$$A \gtrsim B \text{ if } A \ \equiv\ B \text{ and } \forall p \in Pf(A).\ \exists q \in Pf(B).\ p \geq q\ .$$

We define what a *normal-form proof* is, i.e. one of the minimal proofs of $Pf(Th\ A)$:

$$\begin{aligned} Nf(A) &\stackrel{!}{=} \mu Pf(Th\ A) \\ &\stackrel{!}{=} \{p \in Pf(Th\ A)\ :\ \neg\exists q \in Pf(Th\ A).\ p > q\} \end{aligned}$$

The *canonical presentation* contains those formulæ that appear as assumptions of normal-form proofs:

$$A^\sharp \stackrel{!}{=} [Nf(A)]^{Pm}\ .$$

So, we will say that A is *canonical* if $A = A^\sharp$.

A presentation A is *complete* if every theorem has a normal-form proof:

$$Th\ A = [Pf(A) \cap Nf(A)]_{Cl}$$

Canonicity implies completeness, but the converse is not true.

We now consider inference and deduction mechanisms. A *deduction mechanism* $\rightsquigarrow$ is a function from presentations to presentations and we call the relation $A \rightsquigarrow B$ a *deduction step*. A sequence of presentations $A_0 \rightsquigarrow A_1 \rightsquigarrow \cdots$ is called a *derivation*. The *result* of the derivation is, as usual, its *persisting* formulæ:

$$A_\infty \stackrel{!}{=} \liminf_{j\to\infty} A_j = \bigcup_{j>0} \bigcap_{i>j} A_i\ .$$

A deduction mechanism is *completing* if for each step $A \rightsquigarrow B$, $A \gtrsim B$ and the limit A_∞ is complete.

A completing mechanism can be used to build normal-form proofs of theorems of the initial presentation:

Theorem 2 ([4, Lemma 5.13]). *A deduction mechanism is completing if and only if for all derivations $A_0 \rightsquigarrow A_1 \rightsquigarrow \cdots$,*

$$Th\ A_0 \subseteq [Pf(A_\infty) \cap Nf(A_0)]_{Cl}\ .$$

A *critical proof* is a minimal proof which is not in normal form, but whose strict subproofs are:

$$Crit(A) \quad \stackrel{!}{=} \quad \{p \in \mu Pf(A) \setminus Nf(A) : \forall q \in Pf(A).\ p \rhd q \Rightarrow q \in Nf(A)\}$$

Completing formulæ are conclusions of proofs smaller than critical proofs:

$$Comp(A) \quad \stackrel{!}{=} \quad \bigcup_{p \in Crit(A)\ \wedge\ p' \text{ is some proof such that } p > p'} [p']^{Pm}$$

In this paper, we use a completing deduction mechanism in the following way:

$$A \rightsquigarrow A \cup \mathsf{C}(A)$$

where $Comp(A) \subseteq \mathsf{C}(A) \subseteq Th\ A$.

Proposition 2 ([11, Lemma 10]). *This deduction mechanism is completing.*

3 Deduction Modulo is an Instance of ACS

We want to show that the sequent calculus modulo can be seen as an instance of ACS. For this purpose, we have to define what the formulæ, the proofs, the premises and conclusions are, and to give the appropriate orderings. After this, we need to check that the postulates are verified by the defined instance.

3.1 Proofs and Formulae

We aim to obtain cut-free proofs, so that the natural candidate for ACS proofs are sequent calculus proofs. Because of the weakening and contraction properties, we can restrict ourselves to proofs using minimal sets of propositions in their conclusions. More precisely, we can consider only proofs where all the propositions appearing in the conclusion are used as principal propositions somewhere in the proof, or in one of the Axioms.

The completion procedure we want to establish deals with rewrite rules over atomic propositions. Nevertheless, the conclusions of the proofs, from which we want to generate the rewrite rules added by the completion mechanism, are sequents. In other words, sequents must be identified with proposition rewrite systems.

Therefore we suppose that there exists a function between sequents and proposition rewrite systems $Rew : \mathcal{S} \rightarrow \mathcal{PRS}$ such that:

Property 1. For all sequents $\Gamma \vdash \Delta$, $R = Rew(\Gamma \vdash \Delta)$ and $\mathcal{P}(\Gamma \vdash \Delta)$ are strongly compatible :

(a) for all propositions P, Q, $P \xleftrightarrow[R]{*} Q$ implies that there exists a proof of $\mathcal{P}(\Gamma \vdash \Delta) \vdash P \Leftrightarrow Q$ in $\emptyset$ (i.e. without rewrite rules);

(b) there exists a cut-free proof of $\vdash \mathcal{P}(\Gamma \vdash \Delta)$ in R.

Property 2. For all proposition rewrite systems R, for all sequents s and s', if $Rew(s) = Rew(s')$, then s has a proof (resp. cut-free proof) in R iff s' has a proof (resp. cut-free proof) in R.

Property 1 implies compatibility in the sense of Definition 1.4 of [16], which is the same except that we need here a *cut-free* proof in b).

Section 4.3 provides an instance of such a function Rew.

With respect to the definitions of ACSs (see Sect. 2.3) deduction modulo can be seen as an ACS, in the following way:

— $\mathbb{P}$: *proofs* are sequent calculus proofs using minimal sets of propositions in their conclusion:

$$\mathbb{P} \quad \stackrel{!}{=} \quad \{p \in \mathcal{SQM} : \neg\,(\exists q \in \mathcal{SQM}.\ \mathrm{Weak}(q,p))\}$$

where $\mathrm{Weak}(q,p)$ says that the proof p can be obtained from q by weakening.

— $\mathbb{A}$: *formulæ* are proposition rewrite systems corresponding to some sequent:

$$\mathbb{A} \quad \stackrel{!}{=} \quad Rew(\mathcal{S}) \ \subseteq\ \mathcal{PRS}\ .$$

— The *conclusion* of an ACS proof is the rewrite system associated by Rew to the conclusion of the sequent calculus proof: for all proofs p,

$$[p]_{Cl} \quad \stackrel{!}{=} \quad Rew(\Gamma \vdash \Delta) \text{ when } p = \frac{\vdots\ \ \vdots}{\Gamma \vdash \Delta}\ .$$

— The *premises* of a proof are the rewrite systems consisting of the proposition rewrite rules appearing in the proof or its subproofs: for all proofs p,

$$[p]^{Pm} \quad \stackrel{!}{=} \quad \left\{ \left\{ A \to P : \begin{array}{c}\text{there exists a } \uparrow\text{-}\mathsf{l} \text{ or} \\ \uparrow\text{-}\mathsf{r} \text{ using } A \to P \text{ in } q\end{array} \right\} : \begin{array}{l} q \text{ is a subproof of } p \end{array} \right\}$$

This definition implies that we consider only proofs using proposition rewrite systems corresponding to some sequent.

3.2 Orderings on Proofs

We define the following (infinite, but Noetherian) precedence $>$: for all formulæ P, Q, if P is greater than Q for the subformula relation, then $\mathsf{Cut}(P) > \mathsf{Cut}(Q)$, and for all other inference rules r of Fig. 1, $\mathsf{Cut}(P) > \mathsf{r}$.

We order proofs using the RPO [9] based on this precedence. Since the precedence is well-founded, so is the RPO [9]. We restrict this ordering to proofs which have the same *sequent* as conclusion, modulo weakening.

Because we work modulo weakening and contraction, it is important to note that a proof and its weakened and contracted versions are equivalent with respect

to the ordering we have just defined, because they have the same cuts and the same labeled tree structure.

Notice also that with this ordering, a cut-free proof is always strictly smaller than a proof with at least one cut at root.

Subproofs of a proof p are defined as the subproofs of p for the sequent calculus, modulo weakening and contraction (subproofs may use less propositions than their parents).

Unfortunately, this definition is not sufficient to define trivial proofs, because if we use a premise through a $\uparrow$-l or $\uparrow$-r rule, there will always be a strict subproof, so that there is no proofs using premises without strict subproofs.

To solve this problem, we can add manually the trivial proofs, i.e. $\mathbb{P}$ is in fact $\mathbb{P} \cup \widehat{\mathbb{A}}$, where formulæ are identified with their trivial proof.

We have to extend the ordering $>$ to trivial proofs: it can be simply done by saying that they cannot be compared with other proofs. ($>$ over $\mathbb{P} \cup \widehat{\mathbb{A}}$ is the same relation as $>$ over the original $\mathbb{P}$.)

For Postulate C to be verified, we have to extend the subproof relation:

$$p \trianglerighteq q \quad \text{if} \begin{array}{l} - \; q \text{ is a subproof modulo weakening of } p \text{ in } \mathcal{SQM}, \text{ or} \\ - \text{ if } q = \widehat{a} \text{ with } a \in [p]^{Pm}. \end{array}$$

This relation is well-founded because of the wellfoundedness of the subproof relation in sequent calculus, and because trivial proofs cannot have strict subproofs.

With these definitions we can prove the main theorem of this section:

Theorem 3 (Instance of ACS). *The sequent calculus modulo is an instance of ACS, with the definitions of* $\mathbb{A}$, $\mathbb{P}$, $[\cdot]^{Pm}$, $[\cdot]_{Cl}$, $>$ *and* $\rhd$ *given above.*

4 A Generalized Completion Procedure

We want to define a completion procedure through critical proofs. For this, we first need some characterizations of the normal-form proofs and the critical proofs. The limit of this completion procedure will be an equivalent rewrite system which admits Cut.

4.1 Normal-form Proofs and Critical Proofs in Deduction Modulo

Proposition 3 (Characterization of Normal-Form Proofs). *A proof in deduction modulo is in normal form iff it is either a trivial proof or a cut-free proof with no useless logical rules.*

We give now a characterization of the critical proofs in deduction modulo.

Proposition 4 (Critical Proofs in Deduction Modulo). *Critical proofs in deduction modulo are of the form*

$$\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdots\\ \Gamma, A, P \vdash \Delta\end{array}}{\Gamma, A \vdash \Delta}\uparrow\text{-l} \quad \dfrac{\begin{array}{c}\pi'\\ \vdots\\ \Gamma \vdash Q, A, \Delta\end{array}}{\Gamma \vdash A, \Delta}\uparrow\text{-r}}{\Gamma \vdash \Delta}\ \mathsf{Cut}(A)$$

where π and π' are cut-free and do not use unneeded logical rules, and at least one of $A \to P$ or $A \to Q$ is not a term *rewriting.*

Note 1. If we suppose, as in the order condition of [22], that the proposition rewrite system is confluent, and that it is included in a well-founded ordering compatible with the subformula relation, then we can take this ordering instead of the subformula relation to compare cuts in the precedence. Doing this, we can prove that there are no minimal proofs of this form, and consequently *no critical proofs*. Therefore the admissibility of Cut is verified.

The main difference with [22] is that Hermant gives a semantic proof of the cut admissibility, whereas we have here a cut-elimination algorithm, i.e. a terminating syntactical process that transforms a proof into a cut-free one. It remains to be investigated how this process is related with normalization, i.e. β-reduction. (The last case corresponds in fact to an η-expansion.) It is proved in [17] that such an order condition provides normalization in the quantifier-free case.

This result was also independently found by [1], with the same kind of ordering over proofs.

4.2 The Completion Procedure

As we wrote in Sect. 2.3, we want to define a completing deduction mechanism by adding to a presentation A a presentation $\mathsf{C}(A)$ such that $Comp(A) \subseteq \mathsf{C}(A) \subseteq Th\ A$. So we have to find proofs smaller than critical proofs. Here, using Property 1(b) and Lemma 1, we can find for all sequents $\Gamma \vdash \Delta$ a cut-free proof in $Rew(\Gamma \vdash \Delta)$ with conclusion $\Gamma \vdash \Delta$, which will be smaller than any proof containing a cut proving the same sequent, in particular any critical proof. The premises of this proof are in $Rew(\Gamma \vdash \Delta) = [p]_{Cl}$. The best procedure is thus to add only the conclusions of critical proofs. Nevertheless, this is not possible:

Theorem 4 (Undecidability of Critical Proof Search). *Given a propositional rewrite system R and a sequent $\Gamma \vdash \Delta$, it is undecidable to know if $\Gamma \vdash \Delta$ is the conclusion of a critical proof in R.*

We must therefore add a superset of these conclusions. Here we will add the conclusion of the proofs in the form of Proposition 4, except the one that we know for sure that they are not minimal (for instance if $A \in \Gamma \cup \Delta$).

We must consider proofs of the form of Proposition 4. As π and π' are cut-free and do not use unneeded logical rules, they could be found using for instance a

tableaux method modulo, like TaMeD [5], which is complete with respect to cut-free proofs, if we knew Γ and Δ, hence the idea to apply the tableaux method to $A, P \vdash$ and $\vdash Q, A$, and to complete Γ and Δ in order to close the remaining tableaux. Because we work modulo weakening, we can restrict ourselves to the minimal Γ and Δ closing the tableaux. We can then sort the obtained $\Gamma \vdash \Delta$ to remove sequents where $A \in \Gamma \cup \Delta$. The resulting rewrite system is obtained by adding all $Rew(\Gamma \vdash \Delta)$ to our rewrite system.

Theorem 5 (Cut Admissibility of the Limit). *For all sequents $\Gamma \vdash \Delta$, for all proposition rewrite systems R_0, $\Gamma \vdash \Delta$ has a proof in R_0 if and only if it has a* cut-free *proof in R_∞.*

4.3 Sequents and Rewrite Systems

For deduction modulo to be an instance of ACS, we have to define some function *Rew* having Properties 1 and 2. We also want to know how to build proofs that use the rewrite system associated with some sequent, and therefore this function has to be effective.

If we consider only propositional logic (i.e. without quantifiers), we can use the following (non-deterministic) algorithm to transform a set of sequents $\Gamma \vdash \Delta$ into a set of rewrite rules:

Step 1. Choose a sequent. Push all negated formulæ on the other side of the sequent. For instance, $A, \neg B \vdash \neg C, D$ becomes $A, C \vdash B, D$. If the new Γ is empty, go to step 2. If the new Δ is empty, go to step 3. If neither is empty, go to either Step 2 or Step 3.

Step 2. Decompose the last proposition iteratively:

$$\begin{array}{rcl} P_1, \ldots, P_n \vdash Q_1, \ldots, Q_m & \text{becomes} & P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_{m-1} \vdash Q_m \\ P_1, \ldots, P_n \vdash Q_1 \wedge Q_2 & " & P_1, \ldots, P_n \vdash Q_1 \ ; \ P_1, \ldots, P_n \vdash Q_2 \\ P_1, \ldots, P_n \vdash Q_1 \vee Q_2 & " & P_1, \ldots, P_n, \neg Q_1 \vdash Q_2 \\ P_1, \ldots, P_n \vdash Q_1 \Rightarrow Q_2 & " & P_1, \ldots, P_n, Q_1 \vdash Q_2 \\ P_1, \ldots, P_n \vdash A & " & A \rightarrow A \vee \exists x_1, \ldots, x_p.\ (P_1 \wedge \cdots \wedge P_n) \end{array}$$

(A atomic, and the x_i are the free variables appearing in $P_1, \ldots, P_n$ but not in A)

for $P_1, \ldots, P_n \vdash \neg Q$, return to Step 1

Step 3. Decompose the first proposition iteratively, dually from step 2. For instance,

$$\begin{array}{rcl} P_1 \Rightarrow P_2 \vdash Q_1, \ldots, Q_m & \text{becomes} & P_2 \vdash Q_1, \ldots, Q_m \ ; \ \neg P_1 \vdash Q_1, \ldots, Q_m \\ A \vdash Q_1, \ldots, Q_m & " & A \rightarrow A \wedge \forall x_1, \ldots, x_p.\ (Q_1 \vee \cdots \vee Q_m) \end{array}$$

(A atomic, and the x_i are the free variables appearing in $Q_1, \ldots, Q_m$ but not in A)

for $\neg P \vdash Q_1, \ldots, Q_m$, return to Step 1.

This algorithm clearly terminates, because each time a step 2 or 3 begins, either the rewrite rule is generated, or a formula is decomposed into subformulæ, so that the number of connectors different from $\neg$ strictly diminishes. Of course, we do not pretend that this algorithm is the most optimized for our purpose.

$Rew(\Gamma \vdash \Delta)$ will be the function returning the rewrite system obtained by applying the algorithm to $\{\Gamma \vdash \Delta\}$.

This algorithm can be extended to the case with quantifiers. In the case of a $\forall$ on the left of the sequent, or a $\exists$ on the right, we will keep the formula in the sequent, but will not decompose it further. We will therefore denote by $\underline{P}$ the fact that P was already decomposed. Then we do not consider underlined formulæ in Step 1 to choose between Step 2 or Step 3, and at the beginning of Step 2 (resp. Step 3), one keep a non-underlined formula to the right (resp. left) side.
We also have to add the following decomposition steps:

- (2) $P_1, \ldots, P_n \vdash \forall x.\ Q$ becomes $P_1, \ldots, P_n \vdash \{y/x\}Q$ where y does not appear in $P_1, \ldots, P_n$;
- (2) $P_1, \ldots, P_n \vdash \exists x.\ Q$ becomes $P_1, \ldots, P_n, \neg\underline{\exists x.\ Q} \vdash \{t/x\}Q$ where t can be any ground term;
- (3) $\exists x.\ P \vdash Q_1, \ldots, Q_m$ becomes $\{y/x\}P \vdash Q_1, \ldots, Q_m$ where y does not appear in $Q_1, \ldots, Q_m$.
- (3) $\forall x.\ QP_1, \ldots, P_n \vdash$ becomes $\{t/x\}P \vdash \neg\underline{\forall x.\ Q}, Q_1, \ldots, Q_m$ where t can be any ground term.

Of course, at the end, the underlines are removed.

Proposition 5. *The function Rew has the Properties 1 and 2.*

This algorithm does not allow all rewrite systems to be considered as formulæ. Nevertheless, one can transform all rewrite systems to equivalent rewrite systems that are images of sequents by *Rew*, by splitting the rules: $A \to P$ becomes $A \to A \vee P$ and $A \to A \wedge P$. This is equivalent to the polarized rewrite systems of [13].

This algorithm can be seen as the attempt to build a cut-free proof of the conclusion of a critical proof, adding rewrite rules to close the branches were an atomic formula appears.

4.4 Examples

In the case of Crabbé's example presented in the introduction, the input is the rewrite system $A \to B \wedge \neg A$ and the completion procedure generates $B \to B \wedge \bot$ which is equivalent to $B \to \bot$.

With this new rule, we can show that there is no more critical proofs. Therefore, the proposition rewrite system

$$\begin{cases} A \to B \wedge \neg A \\ B \to \bot \end{cases}$$

admits Cut.

The next example deals with quantifiers and is extracted from [22]:

$$R \in R \ \to\ \forall y.\ y \simeq R \Rightarrow y \in R \Rightarrow C$$

where $y \simeq z \quad \stackrel{!}{=} \quad \forall x.\ (y \in x \Rightarrow z \in x)$. It is terminating and confluent, but does not admits Cut.

The critical proofs have the form

$$\dfrac{\dfrac{\vdots}{R \in R, \forall y.\ y \simeq R \Rightarrow y \in R \Rightarrow C \vdash}}{R \in R \vdash} \uparrow\text{-l} \qquad \dfrac{\dfrac{\vdots}{\vdash R \in R, \forall y.\ y \simeq R \Rightarrow y \in R \Rightarrow C}}{\vdash R \in R} \uparrow\text{-l}$$
$$\vdash \qquad \mathsf{Cut}(R \in R)$$

The left part can be developed as

$$\dfrac{\dfrac{\dfrac{R \in R, C \vdash \quad R \in R \vdash t_1 \in R}{R \in R, t_1 \in R \Rightarrow C \vdash} \Rightarrow\text{-l} \qquad \dfrac{\dfrac{R \in R, t_1 \in c_1 \vdash R \in c_1}{R \in R \vdash t_1 \in c_1 \Rightarrow R \in c_1} \Rightarrow\text{-r}}{R \in R \vdash t_1 \simeq R} \forall\text{-r}}{R \in R, t_1 \simeq R \Rightarrow t_1 \in R \Rightarrow C \vdash} \Rightarrow\text{-l}}{R \in R, \forall y.\ y \simeq R \Rightarrow y \in R \Rightarrow C \vdash} \forall\text{-l}$$

and the right part as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{R \in t_0, c_0 \in R \vdash R \in R, C \quad c_0 \in R \vdash c_0 \in t_0, R \in R, C}{c_0 \in t_0 \Rightarrow R \in t_0, c_0 \in R \vdash R \in R, C} \Rightarrow\text{-l}}{c_0 \simeq R, c_0 \in R \vdash R \in R, C} \forall\text{-l}}{c_0 \simeq R \vdash R \in R, c_0 \in R \Rightarrow C} \Rightarrow\text{-r}}{\vdash R \in R, c_0 \simeq R \Rightarrow c_0 \in R \Rightarrow C} \Rightarrow\text{-r}}{\vdash R \in R, \forall y.\ y \simeq R \Rightarrow y \in R \Rightarrow C} \forall\text{-r} \quad .$$

To close the proofs, we can for instance have $t_0 = R = t_1$, and C in the right part of the sequent (to close $R \in R, C \vdash$). One can see that other choices will not produce critical proofs. The resulting sequent is therefore $\vdash C$, and the added rule is $C \to C \vee \top$. This rule does not generate new critical proofs, and consequently, the proposition rewrite system

$$\begin{cases} R \in R \ \to \ \forall y.\ y \simeq R \Rightarrow y \in R \Rightarrow C \\ C \to C \vee \top \end{cases}$$

admits Cut.

One can also think of another example, where there remains quantifiers in the conclusion: consider the following rule derived from Crabbé's example $A \to (\exists x.\ \forall y.\ B \wedge P(x,y)) \wedge \neg A$ where A and B are atomic propositions, and P a predicate of arity 2. For more convenience we will denote by $Q(x)$ the formula $\forall y.\ B \wedge P(x,y)$. We have exactly the same critical proof than in the case of Crabbé, but where B is replaced by $\exists x.\ Q(x)$. The sequent of its conclusion is transformed into a rewrite rule:

$\exists x.\ Q(x) \vdash$ becomes $Q(z) \vdash$
becomes $B \wedge P(z,a) \vdash \neg Q(z)$
becomes $B \vdash \neg P(z,a), \neg Q(z)$
becomes $B \to B \wedge \forall z.\ (\neg P(z,a) \vee \neg Q(z))$.

Then, the resulting systems admits Cut, and a cut free proof of $\exists x.\ Q(x) \vdash$ can be:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q(c), B, P(c,a) \vdash P(c,a)}\,\mathsf{Axiom}}{Q(c), B, \neg P(c,a), P(c,a) \vdash}\,\neg\text{-I} \quad \dfrac{}{Q(c), B, P(c,a) \vdash Q(c)}\,\mathsf{Axiom}}{Q(c), B, \neg P(c,a) \vee \neg Q(c), P(c,a) \vdash}\,\neg\text{-I}}{Q(c), B, \forall z.\ (\neg P(z,a) \vee \neg Q(z)), P(c,a) \vdash}\,\vee\text{-I}}{Q(c), B, B \wedge \forall z.\ (\neg P(z,a) \vee \neg Q(z)), P(c,a) \vdash}\,\forall\text{-I}}{Q(c), B, P(c,a) \vdash}\,\wedge\text{-I}}{Q(c), B \wedge P(c,a) \vdash}\,\uparrow\text{-I}}{Q(c) \vdash}\,\forall\text{-I}}{\exists x.\ Q(x) \vdash}\,\exists\text{-I}$$

It remains to be investigated, as for Knuth-Bendix completion, for which conditions the completion procedure we have defined is terminating. We conjecture that it is the case if the original proposition rewrite system is confluent.

5 Conclusion and Perspectives

We have shown how, by setting the right abstract canonical system structure on the proof space of a sequent calculus modulo, we can use abstract completion to obtain an equivalent theory modulo which deduction admits Cut. This abstract completion is precise enough to be operational, and it is actually implemented. It also reveals an original and deep logical correspondence between the sequent calculus, proof orderings and rewriting completion. This opens several new challenging questions.

The ordering on proofs we are using is adapted to consider cut admissibility as a normal-form property of an ACS, but produces many critical proofs, in particular when quantifiers are involved. This is because some of the rules produced by the completion procedure subsumes others: for instance $A \to A \vee \exists x.\ P(x)$ subsumes $A \to A \vee P(t)$ for a particular $t \in \mathcal{T}(\Sigma, V)$. It is therefore a challenging goal to understand if this ordering could benefit of refinements allowing to target the more relevant critical proofs.

Our saturation procedure only guarantees cut admissibility, not normalization. For instance, with Crabbé's rule, once the system is completed, the initial proof of $B \vdash$ can still be constructed, and it is still not normalizing, i.e. the λ-term that is associated to the proof can be infinitely β-reduced. In other words, we do not have a process that transforms proofs with cuts to cut-free ones. The introduction of simplification rules as in standard completion may allow us to suppress the possibility to build non-normalizing proofs. Moreover, with such simplification rules, the canonical presentation of the system may be obtained.

Let us finally remark that as an interesting consequence of our results, our procedure can be used to determine if a system admits Cut. Indeed, if a proposition rewrite system is a fixpoint of this procedure, then we know that it admits

Cut. The converse is not true, essentially because the procedure uses a superset of the critical proofs. It will be interesting to check what results this procedure will give on systems that are proved to admit Cut, like Higher Order Logic [15] or arithmetic [18].

References

1. Aiguier, M., Boin, C., Longuet, D.: On generalized theorems for normalization of proofs. Technical report, LaMI - CNRS and Université d'Evry Val d'Essonne (2005)
2. Baader, F., Nipkow, T.: Term *Re*writing and *all That*. Cambridge University Press (1998)
3. Barendregt, H., Barendsen, E.: Autarkic computations in formal proofs. Journal of Automated Reasoning **28** (2002) 321–336
4. Bonacina, M.P., Dershowitz, N.: Abstract canonical inference. ACM Trans. Comput. Logic **8** (2007)
5. Bonichon, R.: TaMeD: A tableau method for deduction modulo. In: Basin, D.A., Rusinowitch, M. (eds.): IJCAR. Lecture Notes in Computer Science, Vol. 3097. Springer-Verlag (2004) 445–459
6. Burel, G., Kirchner, C.: Completion is an instance of abstract canonical system inference. In: Futatsugi, K., et al. (eds.): Algebra, Meaning and Computation. Lecture Notes in Computer Science, Vol. 4060. Springer-Verlag (2006) 497–520
7. Burel, G., Kirchner, C.: Cut elimination in deduction modulo by abstract completion (full version). Research report (2007) `http://hal.inria.fr/inria-00132964`.
8. Crabbé, M.: Non-normalisation de la théorie de Zermelo. Manuscript (1974)
9. Dershowitz, N.: Orderings for term-rewriting systems. Theoretical Computer Science **17** (1982) 279–301
10. Dershowitz, N.: Canonicity. In: Dahn, I., Vigneron, L. (eds.): FTP. Electronic Notes in Theoretical Computer Science, Vol. 86. Elsevier Science Publishers B. V. (North-Holland) (2003)
11. Dershowitz, N., Kirchner, C.: Abstract saturation-based inference. In: LICS. IEEE Computer Society (2003) 65–74
12. Dershowitz, N., Kirchner, C.: Abstract Canonical Presentations. Theoretical Computer Science **357** (2006) 53–69
13. Dowek, G.: What is a theory? In: Alt, H., Ferreira, A. (eds.): STACS. Lecture Notes in Computer Science, Vol. 2285. Springer-Verlag (2002) 50–64
14. Dowek, G.: Confluence as a cut elimination property. In: Nieuwenhuis, R. (ed.): RTA. Lecture Notes in Computer Science, Vol. 2706. Springer-Verlag (2003) 2–13
15. Dowek, G., Hardin, T., Kirchner, C.: HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. Mathematical Structures in Computer Science **11** (2001) 1–25
16. Dowek, G., Hardin, T., Kirchner, C.: Theorem proving modulo. Journal of Automated Reasoning **31** (2003) 33–72
17. Dowek, G., Werner, B.: Proof normalization modulo. The Journal of Symbolic Logic **68** (2003) 1289–1316
18. Dowek, G., Werner, B.: Arithmetic as a theory modulo. In: Giesl, J. (ed.): RTA. Lecture Notes in Computer Science, Vol. 3467. Springer-Verlag (2005) 423–437

19. Gallier, J.H.: Logic for Computer Science: Foundations of Automatic Theorem Proving. Computer Science and Technology Series, Vol. 5. Harper & Row, New York (1986) Revised On-Line Version (2003), `http://www.cis.upenn.edu/~jean/gbooks/logic.html`.
20. Gentzen, G.: Untersuchungen über das logische Schliessen. Mathematische Zeitschrift **39** (1934) 176–210, 405–431 Translated in Szabo, editor., *The Collected Papers of Gerhard Gentzen* as "Investigations into Logical Deduction".
21. Hermant, O.: Méthodes Sémantiques en Déduction Modulo. PhD thesis, École Polytechnique (2005)
22. Hermant, O.: Semantic cut elimination in the intuitionistic sequent calculus. In: Urzyczyn, P. (ed.): TLCA. Lecture Notes in Computer Science, Vol. 3461. Springer-Verlag (2005) 221–233
23. Knuth, D.E., Bendix, P.B.: Simple word problems in universal algebras. In: Leech, J. (ed.): Computational Problems in Abstract Algebra. Pergamon Press, Oxford (1970) 263–297
24. Peterson, G., Stickel, M.E.: Complete sets of reductions for some equational theories. Journal of the ACM **28** (1981) 233–264