Cut Admissibility by Saturation

Guillaume Burel

Cut Admissibility by Saturation

Guillaume Burel

ÉNSIIE/Cédric, 1 square de la résistance, 91025 Évry cedex, France
guillaume.burel@ensiie.fr http://www.ensiie.fr/~guillaume.burel/

Abstract. Deduction modulo is a framework in which theories are integrated into proof systems such as natural deduction or sequent calculus by presenting them using rewriting rules. When only terms are rewritten, cut admissibility in those systems is equivalent to the confluence of the rewriting system, as shown by Dowek, RTA 2003, LNCS 2706. This is no longer true when considering rewriting rules involving propositions. In this paper, we show that, in the same way that it is possible to recover confluence using Knuth-Bendix completion, one can regain cut admissibility in the general case using standard saturation techniques. This work relies on a view of proposition rewriting rules as oriented clauses, like term rewriting rules can be seen as oriented equations. This also leads us to introduce an extension of deduction modulo with *conditional* term rewriting rules.

Whatever their origin, proofs rarely need to be searched for without context: Program verification requires arithmetic, theories of lists or arrays, etc. Mathematical theorems are in general not proved in pure predicate logic. Consequently, even if (automated and interactive) proof systems have achieved a high degree of maturity, they need to be able to deal with theories in an efficient way. This explains the particular interest focused on SMT (Satisfiability Modulo Theory) provers in the latter years. However, one of the drawbacks of the SMT approach is that the way theories are integrated is not completely generic, in the sense that each theory needs a special treatment.

A more generic approach to integrating theories into a proof system was proposed by Dowek, Hardin and Kirchner [14]. In Deduction Modulo[1], a theory is represented by a congruence over formulæ, and proofs are searched for modulo this congruence. In practice, this congruence is most often described as a rewriting system. However, using only term rewriting rules would not be enough to capture interesting theories. For instance, Vorobyov [21] showed that even quantifier-free Presburger arithmetic cannot be presented as a convergent term rewriting system. To overcome this, Deduction Modulo also deals with proposition rewriting rules, that rewrite atomic formulæ into formulæ. Thanks to this, it was possible to present many theories in Deduction Modulo: simple type theory (also known as higher-order logic), arithmetic, B set theory [18], any

[1] Although it may sound rather strange, the absence of subsequent to the term "modulo" follows the original works about this field.

pure type system, including the calculus of constructions which is the foundation of the proof assistant Coq [7], or, in fact, any first-order theory [5]. It is then possible to use automated theorem provers based on Deduction Modulo, such as iProver Modulo [6] or Zenon Modulo [9]. Moreover, proofs in those theories can be checked using Dedukti, a proof checker based on Deduction Modulo (https://www.rocq.inria.fr/deducteam/Dedukti/). Note that if one wants that the proof systems modulo a rewriting system behave well, in particular, if one wants the proof search methods to be complete, or the proof calculus to enjoy usual proof-theoretical properties such as the subformula property or the witness property, the rewriting system must have the following feature: The cut rule must be admissible in the sequent calculus modulo the rewriting system. This is true for the presentations of theories cited above.

Even if any first-order theory can be presented as a rewriting system with cut admissibility, these presentations may be quite unnatural. This is particularly the case when equality is involved. Indeed, the work [5] does not handle the equality predicate $\simeq$ in a special way, and for instance an axiom $s \simeq t$ would be presented by a proposition rewriting rule $s \simeq t \to \top$ and not by a term rewriting rule $s \to t$. There are also cases in which the most natural candidate to present an axiom as a rewriting rule would be a conditional rewriting rule, for instance in the case of an axiom of the shape $A(x) \Rightarrow s(x) \simeq t(x)$. In particular, this is the case of one of the axioms of the theory used in the provers of the HOL family (HOL4, HOL Light, or even Isabelle/HOL). In the translation of proofs in the OpenTheory format [17] into proofs that can be checked by Dedukti [1], this axiom could not be easily presented as a rewriting rule, and should therefore remain as an axiom, losing partially the benefit of working modulo the theory. As we will see in Example 8, this axiom can be naturally presented as a conditional rewriting rule. In this paper, we therefore introduce Deduction Modulo Conditional Rewriting Rules, which strictly subsumes the usual presentation of Deduction Modulo.

We therefore need a criterion that ensures that cut admissibility holds in the sequent calculus modulo the conditional rewriting system. To do so, we study links between saturation processes and cut admissibility. In [12], Dowek proved that in the case were there are only *term* rewriting rules, cut admissibility is equivalent to the confluence of the rewriting system. In the case where there are proposition rewriting rules, this is no longer true; for instance the rule $A \to A \Rightarrow B$ is confluent but does not admit cuts. Now, consider a term rewriting system that does not admit cuts. Equivalently, it is not confluent. One way to recover confluence, and thus cut admissibility, is to use the completion technique of Knuth and Bendix [19], that has been refined into Unfailing Completion [2]. Unfailing Completion is a saturation process: starting from a set of equations, new equations are generated, and older ones are simplified, until all newly generated equations are redundant. The set of equations is then called saturated, and in the case of Unfailing Completion, the corresponding rewriting system is convergent on ground terms. Consequently, cut admissibility is ensured for ground terms, which is enough for cut admissibility since we can restrict ourselves to ground sequents (if one considers Eigenvariables as constants). In other words,

when it succeeds, Unfailing Completion allows to recover cut admissibility. In this paper, we investigate how a saturation technique can help at regaining cut admissibility in the more general case when there are *proposition* rewriting rules.

To better apprehend how it works, let us remark that there are usually two ways to see rewriting systems: The first one is to consider them as particular cases of abstract reduction systems whose objects are terms. The second one is to consider them as a set of equations oriented by some reduction ordering. It is this second point of view that is considered in Unfailing Completion, and more generally in the automated theorem proving community. Of course, the two views generally coincide, in particular in the case of terminating rewrite systems. Let us now look at what would correspond to a proposition rewriting rule following the second point of view. According to Dowek [13], a rewriting rule $P \to C$ would coincide to what he calls a one-way clause $\neg P \vee C$, where $\neg P$ is selected, which means it is the only literal that can be used to resolve the clause in the Resolution method. This idea of selected literal is reminiscent of Ordered Resolution with Selection [4], where literals are selected according to a well-founded ordering and a selection function choosing negative literals. Therefore, the analogue of seeing term rewriting rules as equations oriented by an ordering is to see proposition rewriting rules as clauses oriented by an ordering and a selection function. Then, Ordered Resolution with Selection can be used as a saturation process that allows to recover cut admissibility, as we prove in Theorem 7.

We can go a step further. Unfailing Completion and Ordered Resolution with Selection can be combined into Superposition, which is therefore a proof search method for first-order logic with equality. Superposition includes in particular the following inference rule:

$$\mathsf{Superposition}\ \frac{s \simeq u \vee C \qquad L[t]_{\mathfrak{p}} \vee D}{\sigma(L[u]_{\mathfrak{p}} \vee C \vee D)}\ \sigma = mgu(s,t)$$

with ordering restrictions to prevent the proliferation of such inferences. If we look at the inference rule, it behaves as if $L[t]_{\mathfrak{p}}$ was rewritten (or more precisely narrowed) into $L[u]_{\mathfrak{p}}$, provided no condition in C holds. Following our analogy between rewriting rules, equations and clauses, we can therefore see the clause $s \simeq u \vee C$ as a conditional rewriting rule $s \to u$ if $\neg C$. We then prove that when a set of clauses is saturated using Superposition, its corresponding rewriting system, consisting of both proposition rewriting rules and conditional term rewriting rules, admits cuts (Theorem 7 again, since Ordered Resolution with Selection is a special case of Superposition when equality is not present).

In the following section, we will present Deduction Modulo in more details. Then in Section 2 we say a few words about saturation processes, in particular saturation up to compositeness which is a modular form of redundancy. In Section 3 we introduce Deduction Modulo Conditional Rewriting Rules, in particular by means of a sequent calculus. In Section 4, we prove the main result of this paper, namely that when a set of clauses is saturated up to compositeness, then a corresponding conditional rewriting system admits cuts.

1 Deduction Modulo

1.1 Sequent Calculi Modulo

We use standard definitions for terms, predicates, propositions (with connectives $\neg, \Rightarrow, \wedge, \vee$ and quantifiers $\forall, \exists$), sequents, substitutions, term rewriting rules and term rewriting. The substitution of a variable x by a term t in a term or a proposition A is denoted by $\{t/x\}A$, and more generally the application of a substitution σ in a term or a proposition A by σA. A literal is an atomic proposition or the negation of an atomic proposition. The negation of a literal $L^\perp$ is defined by $P^\perp = \neg P$ and $\neg P^\perp = P$. A proposition is in clausal form if it is the universal quantification of a disjunction of literals $\forall x_1, \ldots, x_n.\ L_1 \vee \ldots \vee L_p$ where $x_1, \ldots, x_n$ are the free variables of $L_1, \ldots, L_p$. In the following, we will often omit the quantifiers, and we will identify propositions in clausal form with clauses (i.e. set of literals) as if $\vee$ were associative, commutative and idempotent. This will be justified in Section 3. The symbol $\Box$ represents the empty clause. The polarity of a position in a proposition can be defined as follows: the root is positive, and the polarity switches when going under a $\neg$ or on the left of a $\Rightarrow$.

In deduction modulo, term rewriting is extended to propositions by congruence on the proposition structure. In addition, there are also proposition rewriting rules whose left-hand side is an atomic proposition and whose right-hand side can be any proposition. Such rules can also be applied to non-atomic propositions by congruence on the proposition structure. We call a rewriting system the combination of a term rewriting system and a proposition rewriting system. Deduction modulo consists in applying the inference rules of an existing proof system modulo such a rewriting system.

In this setting, rewriting rules can be applied indifferently to the left- or the right-hand side of a sequent. Consequently, they can be considered semantically as an equivalence between their left- and right-hand sides. To be able to consider implications, a polarized version of deduction modulo was introduced [11]. Proposition rewriting rules are tagged with a polarity $+$ or $-$; they are then called polarized rewriting rules. A proposition A is rewritten positively into a proposition B ($A \longrightarrow^+ B$) if it is rewritten by a positive rule at a positive position or by a negative rule at a negative position. It is rewritten negatively ($A \longrightarrow^- B$) if it is rewritten by a positive rule at a negative position or by a negative rule at a positive position. Intuitively, a positive rule $A \to^+ B$ (resp. a negative rule $B \to^- A$) corresponds to an implication $B \Rightarrow A$. Term rewriting rules (but not proposition rewriting rules) are considered as both positive and negative. $\xrightarrow{*}{}^{\pm}$ is the reflexive transitive closure of $\longrightarrow^{\pm}$. This gives the polarized sequent calculus modulo, some of whose rules are presented in Figure 1.

Example 1. Consider the polarized rewriting system

$$A \subseteq B \to^- \forall x.\ x \in A \Rightarrow x \in B \qquad A \subseteq B \to^+ \neg dw(A,B) \in A$$
$$A \subseteq B \to^+ dw(A,B) \in B$$

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

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

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

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

(dw can be seen as the Skolem symbol introduced by the CNF transformation of the definition of the subset relation, $dw(A, B)$ is a witness that A is not included in B if it is the case.) We can build the following proof of the transitivity of the inclusion in the polarized sequent calculus modulo this system:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{dw(A,C) \in C \vdash A \subseteq C}\ \widehat{\vdash} \qquad \dfrac{}{dw(A,C) \in B \vdash dw(A,C) \in B}\ \widehat{\vdash}}{dw(A,C) \in B \Rightarrow dw(A,C) \in C, dw(A,C) \in B \vdash A \subseteq C}\ \Rightarrow\vdash}{B \subseteq C, dw(A,C) \in B \vdash A \subseteq C}\ \forall\vdash \qquad \dfrac{}{dw(A,C) \in A \vdash dw(A,C) \in A}\ \widehat{\vdash}}{dw(A,C) \in A \Rightarrow dw(A,C) \in B, B \subseteq C, dw(A,C) \in A \vdash A \subseteq C}\ \Rightarrow\vdash}{A \subseteq B, B \subseteq C, dw(A,C) \in A \vdash A \subseteq C}\ \forall\vdash}{A \subseteq B, B \subseteq C \vdash A \subseteq C, A \subseteq C}\ \vdash\neg}{A \subseteq B, B \subseteq C \vdash A \subseteq C}\ \vdash\vee$$

We denote by $\Gamma \vdash_{\mathcal{R}} \Delta$ the fact that the sequent $\Gamma \vdash \Delta$ is provable in the Polarized Sequent Calculus Modulo $\mathcal{R}$. A theory Γ and a rewriting system $\mathcal{R}$ are called *compatible* if for all formulæ A, then $\Gamma \vdash A$ (without rewriting) if and only if $\vdash_{\mathcal{R}} A$.

The cut rule is admissible in the sequent calculus modulo $\mathcal{R}$ if, whenever a sequent can be proved in it, then it can be proved without using the cut rule ($\overset{\smile}{\vdash}$ in Figure 1). Abusing terminology, we say that a rewriting system $\mathcal{R}$ admits cut if the cut rule is admissible in the sequent calculus modulo $\mathcal{R}$. The admissibility of the cut rule has a strong proof-theoretical as well as practical importance: it entails that normal forms exist for proofs; it implies the consistency of the theory associated to $\mathcal{R}$; it is equivalent to the completeness of the proof search procedures based on deduction modulo $\mathcal{R}$; etc.

1.2 Resolution Modulo and One-Way Clauses

An extension of resolution based on deduction modulo, called ENAR for Extended Narrowing and Resolution, was proposed by Dowek, Hardin and Kirchner [14]. It consists of adding a new inference rule to the method of Robinson [20]. This rule, called Extended Narrowing, narrows a clause using the rewrite system modulo which the proof is searched for. However, in ENAR, there is a need to

$$\textsf{Resolution}\ \frac{P \vee C \qquad \neg Q \vee D}{\sigma(C \vee D)}\ \sigma = mgu(P,Q) \qquad \textsf{Factoring}\ \frac{L \vee K \vee C}{\sigma(L \vee C)}\ \sigma = mgu(L,K)$$

$$\textsf{Ext. Narr.}^{-}\ \frac{P \vee C}{\sigma(D \vee C)}\ \begin{smallmatrix}\sigma = mgu(P,Q)\\ Q \to^{-} D \in \mathcal{R}\end{smallmatrix} \qquad \textsf{Ext. Narr.}^{+}\ \frac{\neg Q \vee D}{\sigma(C \vee D)}\ \begin{smallmatrix}\sigma = mgu(P,Q)\\ P \to^{+} \neg C \in \mathcal{R}\end{smallmatrix}$$

Fig. 2. Polarized Resolution Modulo

transform formulæ into clausal normal form during proof search, and not only before as it is usually the case with resolution methods. Therefore, Dowek refined ENAR into Polarized Resolution Modulo, whose rules are presented in Figure 2. In Polarized Resolution Modulo, proposition rewrite rules are assumed to be *clausal*, which means that positive rewrite rules are of the form $P \to^{+} \neg C$, and negative rules are of the form $P \to^{-} C$, where C is in clausal form. This ensures that formulæ generated by Extended Narrowing are still in clausal form.

Applying Extended Narrowing to a clause $P \vee C$ using the rule $Q \to^{-} D$ produces the same clause (namely $\sigma(D \vee C)$, where $\sigma = mgu(P,Q)$) as applying Resolution to this clause $P \vee C$ and the clause $\neg Q \vee D$. Similarly, narrowing with $P \to^{+} \neg C$ amounts to resolving with $P \vee C$. Therefore, the polarized rewrite rule $Q \to^{-} D$ (resp. $P \to^{+} \neg C$) can be identified with what Dowek [13] called the one-way clause $\underline{\neg Q} \vee D$ (resp. $\underline{P} \vee C$) where

- two one-way clauses cannot be resolved together;
- only the selected (underlined) literal of a one way-clause can be used in resolution.

Conversely, given a clause C and a literal L in C, it is always possible to associate a polarized rewrite rule $\mathit{polar}(C,L)$: $\mathit{polar}(P \vee C, P)$ is $P \to^{+} \neg C$ and $\mathit{polar}(\neg Q \vee D, \neg Q)$ is $Q \to^{-} D$. Therefore, the same way that it is possible to see a term rewriting rule as an equation in which one side is selected, it is possible to see clausal polarized rewriting rules as clauses in which a literal is selected.

2 Saturation

If there are only term rewriting rules in $\mathcal{R}$, and no proposition rewriting rules, Dowek [12] showed that cut admissibility in the asymmetric sequent calculus modulo $\mathcal{R}$ is equivalent to confluence of $\mathcal{R}$. If $\mathcal{R}$ is not confluent, a way to get an equivalent rewriting system which is confluent is to apply the Knuth-Bendix standard completion [19], which was extended into Unfailing Completion [2]. Unfailing completion can be seen as a saturation process: one applies all possible inferences to a starting set of formulæ (in that case, positive unit equations) until all newly inferred formulæ are redundant, that is, can be simplified. In that case, the resulting set is called saturated, and the correctness of the procedure shows that a saturated set has the required property, namely ground convergence in the case of Unfailing Completion. Of course, since the required property is in

general not decidable, the saturation process may not terminate. Resolution and its refinements can also be seen as saturation processes: the set of clauses is completed until either the empty clause is generated, in which case the initial set was inconsistent, or until all newly generated clauses are redundant, in which case it is possible to construct a model of the saturated set of clauses.

It would be preferable that the saturation process were modular, in the sense that, if a set Γ of formulæ is saturated, then saturating $\Gamma \cup \Delta$ should not need to apply inferences between formulæ of Γ only. (This is in particular crucial for implementing resolution using the given clause algorithm, to ensure that clauses that were redundant remains redundant when a new given clause is chosen.) Therefore, redundancy should be modular, in the sense that if C is redundant in Γ, then it should be redundant in $\Gamma \cup \Delta$. This refinement of redundancy is called compositeness by Bachmair and Ganziger [3]. In fact, resolution-based provers saturate their input in general not up to redundancy but up to compositeness (Bachmair and Ganziger call saturation up to compositeness completeness, but we will keep writing "saturation up to compositeness" to keep things clear.) Of course, saturation up to compositeness implies saturation up to redundancy.

To deal with full first-order logic with equality, and not only unit clauses, Unfailing Completion can be extended into Superposition [3], which is consequently a complete proof-search method for first-order logic with equality. In pure Superposition, the only predicate is the equality predicate (noted $\simeq$), and clauses are therefore sets of equations and inequations. It is possible to encode other predicates using function symbols, as is done for instance in the prover E. However, to separate more clearly reasoning about equality and about propositions, we will use the inference for Superposition in addition to the rules for Ordered Resolution with Selection [4] (a refinement of resolution inspired by Superposition), as is done in the prover SPASS. As in Unfailing Completion, we consider a reduction ordering $\succ$, that is, an ordering that is stable under substitution and context. Literals are compared as the multisets of multisets $\{\{s\}, \{t\}\}$ for the positive literal $s \simeq t$ and $\{\{s, \flat\}, \{t, \flat\}\}$ for the negative literal $s \not\simeq t$, where $\flat$ is a special symbol not part as the signature, which is assumed to be smaller than any term. A clause $s \simeq t \vee C$ is *reductive* for $s \simeq t$ if $t \not\succeq s$ and $s \simeq t$ is strictly maximal in $s \simeq t \vee C$. We also consider a selection function S that, given a clause C, returns a subset of the negative literals of C. Without considering simplifications, Superposition consists of the four inference rules presented in Figure 3, in addition to which we also consider the two rules of Ordered Resolution with Selection presented in Figure 4.

As we can see, Superposition has strong restrictions on the application of inference rules, which explain in part its efficiency. In particular, ordering restrictions are performed after the application of the unifier σ. Let us note notwithstanding that, thanks to the stability of $\succ$ by substitution, the calculus remains of course complete if the restriction is applied on the premises, although it makes the proof search space bigger.

In Superposition, compositeness can be defined as follows: A ground clause C is called composite with respect to Γ if there exists ground instances $C_1, \ldots, C_n$

$$\textsf{Equality Resolution}\ \frac{s \not\simeq t \vee C}{\sigma(C)} \qquad \textsf{Negative Superposition}\ \frac{s \simeq u \vee C \qquad \neg P[t]_{\mathfrak{p}} \vee D}{\sigma(\neg P[u]_{\mathfrak{p}} \vee C \vee D)}$$

$$\textsf{Positive Superposition}\ \frac{s \simeq u \vee C \qquad P[t]_{\mathfrak{p}} \vee D}{\sigma(P[u]_{\mathfrak{p}} \vee C \vee D)} \qquad \textsf{Eq. Factoring}\ \frac{s \simeq u \vee t \simeq v \vee C}{\sigma(u \not\simeq v \vee t \simeq v \vee C)}$$

where

1. in all rules above, $\sigma = mgu(s,t)$;
2. in Equality Resolution, either $s \not\simeq t \in S(s \simeq u \vee C)$, or $(S(s \simeq u \vee C) = \emptyset$ and $\sigma(s \not\simeq t)$ is maximal in $\sigma(s \simeq u \vee C)$;
3. in both Superpositions, $\sigma(s \simeq u \vee C)$ is reductive for $\sigma(s \simeq u)$ and t is not a variable;
4. in Negative Superposition, either $\neg P[t]_{\mathfrak{p}} \in S(\neg P[t]_{\mathfrak{p}}, D)$ or $S(\neg P[t]_{\mathfrak{p}} \vee D) = \emptyset$ and $\sigma(\neg P[t]_{\mathfrak{p}})$ is maximal in $\sigma(\neg P[t]_{\mathfrak{p}} \vee D)$;
5. in Positive Superposition, $S(P[t]_{\mathfrak{p}} \vee D) = \emptyset$ and $\sigma(P[t]_{\mathfrak{p}} \vee D)$ is reductive for $\sigma(P[t]_{\mathfrak{p}})$;
6. in Equality Factoring, $S(s \simeq u \vee t \simeq v \vee C) = \emptyset$ and $\sigma(s \simeq u)$ is maximal in $\sigma(s \simeq u \vee t \simeq v \vee C)$;
7. in both Superpositions, if $P[t]_{\mathfrak{p}}$ is an equation $v[t]_{\mathfrak{p}'} \simeq w$, then $\sigma w \not\succeq \sigma v[t]_{\mathfrak{p}'}$.

Fig. 3. Inference Rules of Superposition

$$\textsf{Resolution}\ \frac{P \vee C \qquad \neg Q \vee D}{\sigma(C \vee D)} \qquad\qquad \textsf{Factoring}\ \frac{P \vee Q \vee C}{\sigma(P \vee C)}$$

where

1. in both cases, $\sigma = mgu(P,Q)$;
2. in Resolution, σP is strictly maximal in $\sigma(P \vee C)$ and $S(\sigma(P \vee C)) = \emptyset$;
3. in Resolution, either $\sigma(\neg Q) \in S(\sigma(\neg Q \vee D)$ or $S(\sigma(\neg Q \vee D) = \emptyset$ and $\sigma(\neg Q)$ is maximal in $\sigma(\neg Q \vee D)$;
4. in Factoring, σP is maximal in $\sigma(P \vee C)$ and $S(\sigma(P \vee C)) = \emptyset$.

Fig. 4. Inference Rules of Ordered Resolution with Selection

of clauses of Γ such that $C_1, \ldots, C_n$ entails C and $C \succ C_j$ for all $1 \leq j \leq n$. A non-ground clause is called composite with respect to Γ if all its ground instances are. Lemma 11 of [3] tells us that if C is composite in Γ, then it is composite in $\Gamma \cup \Delta$, and that all composite clauses can be safely removed from Γ, as expected.

3 Deduction Modulo Conditional Rewriting Rules

We now present an extension of deduction modulo to the case of conditional rewriting rules.

Definition 2 (Conditional rewriting rule). *A conditional rewriting rule is given by a pair of terms t and s and a set of formalæ Γ. It is denoted by $s \to t$ if Γ.*

A term u rewrites to a term v and conditions Δ using the conditional rewriting rule $s \to t$ if Γ if there is a position $\mathfrak{p}$ and a substitution θ such that $u_{|\mathfrak{p}} = \theta s$,

$v = u[\theta t]_{\mathfrak{p}}$ and $\Delta = \theta\Gamma$. This rewrite relation is denoted by $u \longrightarrow v \wr \Delta$. This is extended to propositions by congruence on the proposition structure.

Note that our definition of conditional rewriting differs from the usual definition as can be found in [10], because conditions are not checked before applying the rewriting rule, but they are delayed and, as we will see, they are checked using a proof system and not just using normalization. This allows to have arbitrary first-order formulæ as conditions.

We can combine conditional (term) rewriting rules $\mathcal{C}$ with polarized (proposition) rewriting rules $\mathcal{P}$ to get what we call a polarized and conditional rewriting system : A proposition A rewrites positively to B and Δ in $\mathcal{CP}$ ($A \xrightarrow[\mathcal{CP}]{*} {}^{+}B \wr \Delta$) if

- either $A = B$ and $\Delta = \emptyset$
- or $A \xrightarrow[\mathcal{CP}]{*} {}^{+}A' \wr \Delta_1$ and
  - either $A' \xrightarrow[\mathcal{P}]{} {}^{+}B$ and $\Delta = \Delta_1$
  - or $A' \xrightarrow[\mathcal{C}]{} B \wr \Delta_2$ and $\Delta = \theta\Delta_1 \cup \Delta_2$ where θ is a renaming of the free variables of Δ_1 to avoid clashes with those of Δ_2.

Negative rewriting can be defined similarly. Let us remark that a conditional rewriting step is therefore both a positive or a negative step.

Definition 3. *A formula* does not involve equality *if $\simeq$ is not present in it. A conditional rewriting rule $s \to t$ if C* does not involve equality *if all formulæ in C do not involve equality.*

Given a rewriting system that does not involve equality, we can define the Sequent Calculus Modulo Polarized and Conditional Rules. We only give the main rules in Figure 5, the others can be induced from them:

$$\vdash\subset \frac{\Gamma \vdash \theta A_1, \Delta \quad \dots \quad \Gamma \vdash \theta A_n, \Delta}{\Gamma, A \vdash B, \Delta} \; A \xrightarrow[\mathcal{CP}]{*} {}^{-}C \wr \{A_1; \dots; A_i\} \text{ and } B \xrightarrow[\mathcal{CP}]{*} {}^{+}C \wr \{A_{i+1}; \dots; A_n\}$$

$$\subset\vdash \frac{\Gamma, A \vdash \Delta \quad \Gamma \vdash B, \Delta \quad \Gamma \vdash \theta A_1, \Delta \quad \dots \quad \Gamma \vdash \theta A_n, \Delta}{\Gamma \vdash \Delta} \; \begin{array}{l} C \xrightarrow[\mathcal{CP}]{*} {}^{-}A \wr \{A_1; \dots; A_i\} \\ C \xrightarrow[\mathcal{CP}]{*} {}^{+}B\{A_{i+1}; \dots; A_n\} \end{array}$$

$$\Rightarrow\vdash \frac{\Gamma, B \vdash \Delta \quad \Gamma \vdash A, \Delta \quad \Gamma \vdash \theta A_1, \Delta \quad \dots \quad \Gamma \vdash \theta A_n, \Delta}{\Gamma, C \vdash \Delta} \; C \xrightarrow[\mathcal{CP}]{*} {}^{-}A \Rightarrow B \wr \{A_1; \dots; A_n\}$$

$$\vdash\top \frac{\Gamma \vdash \theta A_1, \Delta \quad \dots \quad \Gamma \vdash \theta A_n, \Delta}{\Gamma \vdash A, \Delta} \; A \xrightarrow[\mathcal{CP}]{*} {}^{+}\top \wr \{A_1; \dots; A_n\}$$

$$\forall\vdash \frac{\Gamma, \{t/x\}A \vdash \Delta \qquad \Gamma \vdash \theta A_1, \Delta \quad \dots \quad \Gamma \vdash \theta A_n, \Delta}{\Gamma, B \vdash \Delta} \; B \xrightarrow[\mathcal{CP}]{*} {}^{-}\forall x.\ A \wr \{A_1; \dots; A_n\}$$

where θ is a substitution of the free variables of all A_i.

Fig. 5. Some Inference Rules of the Sequent Calculus Modulo Polarized and Conditional Rules

Example 4. Consider the following polarized and conditional rewriting system, inspired by Collatz conjecture:

$$\begin{array}{rl@{\qquad}rl}
syracuse(X) & \to syracuse(half(X)) \text{ if } \{even(X)\} & half(s(o)) & \to o \text{ if } \emptyset \\
syracuse(X) & \to syracuse(tnpo(X)) \text{ if } \{odd(X)\} & half(o) & \to o \text{ if } \emptyset \\
half(s(s(X)) & \to s(half(X)) \text{ if } \emptyset & even(s(X)) & \to^+ odd(x) \\
tnpo(s(X)) & \to s(s(s(tnpo(X)))) \text{ if } \emptyset & odd(s(X)) & \to^+ even(x) \\
tnpo(o) & \to s(o) \text{ if } \emptyset & even(o) & \to^+ \top
\end{array}$$

and let us denote by $\underline{n}$ the term $\underbrace{s(\dots s(}_{n \text{ times}} o))$.

Then, $syracuse(\underline{5}) \xrightarrow{*} \underline{1} \wr \{odd(\underline{5}); even(\underline{16}); even(\underline{8}); even(\underline{4}); even(\underline{2})\}$, and we have the following proof of $odd(syracuse(\underline{5}))$:

$$\dfrac{\vdash\top \dfrac{\vdash\top \dfrac{}{\vdash odd(\underline{5})}}{\vdash even(\underline{8})} \quad \vdash\top \dfrac{\vdash\top \dfrac{}{\vdash even(\underline{16})}}{\vdash even(\underline{4})} \quad \vdash\top \dfrac{}{\vdash even(\underline{2})}}{\vdash odd(syracuse(\underline{5}))}$$

Note that Deduction Modulo Polarized and Conditional Rules strictly subsumes Polarized Deduction Modulo, which is exactly the case when there are no conditions in the term rewriting rules.

Lemma 5. *The inference rules $\forall\vdash$, $\vee\vdash$, $\neg\vdash$ and $\vdash\neg$ are invertible in the Sequent Calculus Modulo Polarized and Conditional Rules, which means that their conclusion is provable if and only if their premises are. Moreover, if the proof of the conclusion does not use $\vdash\!\!\!\smile$, neither do the proofs of the premises.*

Proof. By induction on the proof of the conclusion. □

This lemma implies that we can handle formulæ in clausal normal form as clauses, that is, set of literals. See for instance [16].

4 Ensuring Cut Admissibility using Saturation

Given an ordering $\succ$ and a selection function S, we define the polarized and conditional rewriting system associated to a set of clauses, and state that the saturation of the set of clauses implies the cut admissibility for the corresponding rewriting system.

Definition 6. *Given a set of clauses Γ, an ordering on terms $\succ$ and a selection function S, then the rewrite system $\mathcal{CS}(\Gamma, \succ, S)$ contains all rewrites rules*

- *$polar(C, L)$ for all clauses C such that $S(C) \neq \emptyset$ and for all L in $S(C)$ that does not involve equality;*
- *$polar(C, L)$ for all clauses C such that $S(C) = \emptyset$ and for all L maximal in C w.r.t. $\succ$ that does not involve equality;*

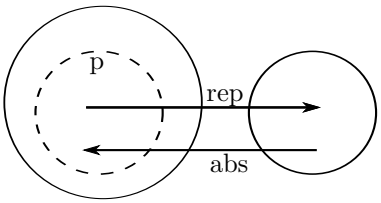
– $s \to t$ *if* $\{L_1^\perp, \ldots, L_n^\perp\}$ *for all clauses* $C = s \simeq t \vee L_1 \vee \cdots \vee L_n$ *such that* $S(C) = \emptyset$, $s \simeq t$ *is maximal in* C *and* $t \not\succ s$.

Theorem 7. *If the set* Γ *of clauses is saturated by Superposition up to compositeness, and* $\mathcal{CS}(\Gamma, \succ, S)$ *does not involve equality, then the sequent calculus modulo* $\mathcal{CS}(\Gamma, \succ, S)$ *is compatible with* Γ *and it admits cuts.*

Note that the Sequent Calculus Modulo Polarized and Conditional Rules is only defined to prove formulæ that do not involve equality.

Before we prove Theorem 7, let us look at an example.

Example 8. In provers of the HOL family, it is possible to define a new type corresponding to the (non-empty) set of terms that satisfies a predicate p. To do so, two function symbols *abs* and *rep* are introduced that go respectively from the initial type to the new one and conversely, as is represented in the following figure:



These function symbols satisfy the axioms $\forall X.\ p(X) \Leftrightarrow abs(rep(X)) \simeq X$ and $\forall Y.\ rep(abs(Y)) \simeq Y$, which correspond to the clauses

$$p(X) \vee \neg abs(rep(X)) \simeq X \tag{1}$$
$$\neg p(X) \vee abs(rep(X)) \simeq X \tag{2}$$
$$rep(abs(Y)) \simeq Y \tag{3}$$

Without a selection function, and with the lexicographic path ordering with precedence $abs \succ p$ and $rep \succ p$, the resulting conditional rewriting system is

$$abs(rep(X)) \to X \text{ if } \{p(X)\} \qquad rep(abs(Y)) \to Y \text{ if } \emptyset$$

The sequent calculus modulo this system is not compatible with the initial theory. Indeed, it is not possible to prove $\forall Y.\ p(abs(Y))$, although this is a consequence of the axioms. This comes from the fact that the set of clauses is not saturated for Superposition. To saturate the set of clause, for instance using E, we only need to add a new clause, namely $p(abs(Y))$, obtained by applying Negative Superposition on (3) and (1), and then Equality Resolution on $p(abs(X)) \vee \neg abs(X) \simeq abs(X)$, which is then composite. Note that all other generated clauses are tautologies, and therefore are composite.

Consequently, the polarized and conditional rewriting system

$$abs(rep(X)) \to X \text{ if } \{p(X)\} \quad rep(abs(Y)) \to Y \text{ if } \emptyset \quad p(abs(X)) \to^+ \neg\bot$$

admits cut for formulæ that do not involve equality.

Let us mention Holide [1], a translator of proofs in the OpenTheory [17] format into Dedukti, a proof checker based on deduction modulo. Most of the theory of HOL can be expressed as rewriting rules, except a few axioms that cannot be easily oriented. The first axiom defining *rep* and *abs* is one of these axioms, and as we have seen, it could be oriented as a conditional rewriting rule.

To prove Theorem 7, we need to prove that if A is proved in the sequent calculus modulo $\mathcal{CS}(\Gamma, \succ, S)$, then it can be proved in Γ, and that if A can be proved in Γ, then it can be proved in the sequent calculus modulo $\mathcal{CS}(\Gamma, \succ, S)$ without $\vdash\!\!\!\smile$. The proof is partially based on the work of Dowek [13] who proves that a derivation in PRM can be translated into a cut-free proof in the polarized sequent calculus modulo.

Lemma 9. *If* $A \xrightarrow{*} {}^{-}B \wr \{L_1, \ldots, L_n\}$, *then* $\Gamma, A \vdash B, L_1^{\perp}, \ldots, L_n^{\perp}$.
If $A \xrightarrow{*} {}^{+}B \wr \{L_1, \ldots, L_n\}$, *then* $\Gamma, B \vdash A, L_1^{\perp}, \ldots, L_n^{\perp}$.

Proof. By induction on the length of the derivation; several steps can be combined using $\vdash\!\!\!\smile$. Note the importance of renaming free variables between several steps. A single step can be proved by induction on the rewritten formula. If the rewriting occurs in a subformula, we can use the induction hypothesis to conclude. Let us therefore assume that A is atomic. We have two cases depending on whether a polarized or a conditional rule is used.

- $A \longrightarrow B \wr \{L_1, \ldots, L_n\}$. There is a rule $s \to t$ if $\{L'_1, \ldots, L'_n\}$ in $\mathcal{CS}(\Gamma, \succ, S)$ and a substitution σ such that $A_{|\mathfrak{p}} = \sigma s$, $B = A[\sigma t]_{\mathfrak{p}}$ and $L_k = \sigma L'_k$. This rule corresponds to a clause $s \simeq t \vee {L'_1}^{\perp} \vee \cdots \vee {L'_n}^{\perp}$ in Γ. From Γ, A one can therefore deduce $B \vee {L_1}^{\perp} \vee {L_n}^{\perp}$, and thus $\Gamma, A \vdash B, L_1^{\perp}, \ldots, L_n^{\perp}$.
- $A \longrightarrow {}^{-}B$. There exists a rule $P \to^{-} C$ in $\mathcal{CS}(\Gamma, \succ, S)$ and a substitution σ such that $A_{|\mathfrak{p}} = \sigma P$ and $B = \sigma C$. Therefore, there is a clause $\neg P \vee C$ in Γ, and from Γ, A one can therefore deduce σC, thus $\Gamma, A \vdash B$. □

Corollary 10. *If* $\Pi \vdash_{\mathcal{CS}(\Gamma, \succ, S)} \Delta$ *then* $\Gamma, \Pi \vdash \Delta$.

Proof. By induction on the proof, using Lemma 9 to convert rewriting steps. □

Lemma 11. *If the set* Γ *of clauses is saturated by Superposition up to compositeness, and* $\mathcal{CS}(\Gamma, \succ, S)$ *does not involve equality, if* A *does not involve equality and* A *is valid in* Γ, *then it can be proved in the sequent calculus modulo* $\mathcal{CS}(\Gamma, \succ, S)$ *without* $\vdash\!\!\!\smile$.

Proof. Since Superposition is complete, there is a derivation of the empty clause from Γ and $\mathscr{C}(\neg A)$ (the clausal normal form of $\neg A$). We are going to translate this derivation into a cut-free proof of $\mathscr{C}(\neg A) \vdash$, by induction of the length of the derivation.

Let us show that all new clauses in the derivation of □ do not involve equality: Since Γ is saturated up to compositeness, all inferences using only premises in Γ are redundant and therefore can be discarded. So by induction hypothesis at least one of the premises does not involve equality. The only way to obtain an

equality would be to apply Resolution or Superposition on a literal L of a clause $L \vee C$ of Γ. In the former case, the restriction on the application of Resolution implies that $\mathit{polar}(C, L)$ is in $\mathcal{CS}(\Gamma, \succ, S)$. Because it does not involve equality, this means that C, and thus the new clause, neither do. In the latter case, the clause involving equality is necessarily the left one in the Superposition inference rule, so that $L = s \simeq t$ for some s and t, and $C = L_1 \vee \cdots \vee L_n$. The restriction on the application of Superposition implies that $s \to t$ if $\{L_1^\perp, \ldots, L_n^\perp\}$ is in $\mathcal{CS}(\Gamma, \succ, S)$, so that $L_1, \ldots, L_n$ do not involve equality, and thus the new clause neither.

Therefore, since Γ is saturated up to compositeness, we can assume that the derivation of $\square$ does not contain applications of Equality Resolution or Equality Factoring. Let us look at the remaining cases. To ease the proof, we can decompose the application of the inference rules into the application of an instantiation and the application of the rule without unification, as in the PEIR calculus of [13]. We have the following cases:

- Instantiation of a clause C outside Γ into σC. By induction hypothesis we have a cut-free proof of $\Delta, C, \sigma C \vdash$. We can build a cut-free proof of $\Delta, C \vdash$ by applying a contraction of C and then $\forall\vdash$ to instantiate the variables as in σ. (Remind that we omit to write quantifiers in clauses, so that C stands in fact for $\forall x_1, \ldots, x_n, C$ where $x_1, \ldots, x_n$ are the free variable of C.)
- Resolution between two clauses $P \vee C$ and $\neg P \vee D$ outside Γ. Let us suppose that we have a cut-free proof of $\Delta, P \vee C, \neg P \vee D, C \vee D \vdash$, then Proposition 7 of [13] implies that we have a proof of $\Delta, P \vee C, \neg P \vee D \vdash$.
- Resolution between a clause $P \vee C$ outside Γ and a clause obtained by instantiating a clause $\neg Q \vee D$ of Γ with substitution σ. Then $Q \to^- D$ is in $\mathcal{CS}(\Gamma, \succ, S)$, and $P \xrightarrow{*} {}^-\sigma D \wr \emptyset$. By induction hypothesis, we have a cut-free proof of $\Delta, P \vee C, C \vee \sigma D \vdash$. We can obtain a cut-free proof of $\Delta, P \vee C \vdash$ by applying a contraction of $P \vee C$ and rewriting P into σD.
- Resolution between a clause $\neg P \vee C$ outside Γ and a clause obtained by instantiating a clause $Q \vee D$ of Γ: similar to the previous case, except that we need to eliminate a double negation.
- Superposition between an clause obtained by instantiating a clause $s \simeq t \vee L_1 \vee \cdots \vee L_n$ of Γ with substitution σ and a clause $L[\sigma s]_\mathfrak{p} \vee D$ outside Γ. The restriction on the application of Superposition implies that the rule $s \to t$ if $\{L_1^\perp, \ldots, L_n^\perp\}$ is in $\mathcal{CS}(\Gamma, \succ, S)$, and consequently $L[\sigma s]_\mathfrak{p} \xrightarrow{*} L[\sigma t]_\mathfrak{p} \wr \{\sigma L_1^\perp, \ldots, \sigma L_n^\perp\}$. By induction hypothesis, we have a cut-free proof of Δ, $L[\sigma s]_\mathfrak{p} \vee D, L[\sigma t]_\mathfrak{p} \vee D \vee L_1 \vee \cdots \vee L_n \vdash$. Since $\vee\vdash$ is cut-free invertible, we therefore have (cut-free) proofs of $\Delta, L[\sigma s]_\mathfrak{p} \vee D, L[\sigma t]_\mathfrak{p} \vee D \vdash$ and $\Delta, L[\sigma s]_\mathfrak{p} \vee D, L_i \vdash$ for all $1 \leq i \leq n$. Starting from $\Delta, L[\sigma s]_\mathfrak{p} \vee D \vdash$, we can apply contraction and rewrite $L[\sigma s]_\mathfrak{p}$, so that it remains to prove $\Delta, L[\sigma s]_\mathfrak{p} \vee D, L[\sigma t]_\mathfrak{p} \vee D \vdash$, which we have, and $\Delta, L[\sigma s]_\mathfrak{p} \vee D \vdash \sigma {L_i}^\perp$ for all $1 \leq i \leq n$, which can be obtained by application of $\vdash\neg$ or inversion of $\neg\vdash$ in the proofs above.
  Note that to make the proof more clear, we did not take the universal quantifiers into account: this is not a problem since $\forall\vdash$ is invertible, and we took care of renaming the free variables of the conditions during rewriting.

The last point to show is that from a cut-free proof of $\mathscr{C}(\neg A) \vdash$ one can build a cut-free proof of $\vdash A$. This can be proved by slightly adapting the proof of Proposition 3 of [16]. □

5 Conclusion and Further Works

We have introduced an extension of Deduction Modulo that handles *conditional* rewriting rules. To get a criterion for cut admissibility in that setting, we have examined how rewriting rules can be seen as oriented equations and oriented clauses. This reflection has lead us to study how saturation techniques can help presenting a theory through a rewriting system with cut admissibility. Our main result is that whenever a set of clauses is saturated, we can build a corresponding rewriting system admitting cuts. We can therefore use state-of-the-art automated theorem provers, which are based on saturation techniques, to orient a theory so that it can be used in Deduction Modulo. These notable results could be extended in several directions.

First, the conditions in the conditional rewriting rules obtained from a saturated set of clauses are simple, since they are only a set of literals. This comes from the fact that we start from clauses, and not arbitrary formulæ. To get more interesting conditions, an idea would be to consider the work of Ganzinger and Stuber [15] that extend Superposition with formulæ that need not be in clausal normal form.

Second, our work is restricted to the case where equality appears only in the rewriting rules, not in the conditions nor in the formulæ to be proved. If we allowed equations in them, Negative Superposition could be applied to clauses of the theory in which a negative equation is selected. Therefore, these clauses could not be discarded as it is the case in Definition 6. Another issue would be the design of a sequent calculus modulo for first-order logic with equality. It could be handled by extending one of the calculi of [8].

Third, saturation implies cut admissibility, but the converse is not true in general. It would be interesting to be able to characterize cut admissibility as precisely as can be done when only terms are rewritten, where it is equivalent to the well studied notion of confluence.

Finally, it would be interesting to see how our work on conditional rewriting rules can be extended to the $\lambda\Pi$-calculus modulo, the system at the heart of the proof checker Dedukti. By doing so, we would be able to orient the theory used in the provers of the HOL family without using axioms, thus improving the performance of the translator Holide [1].

References

1. Assaf, A., Burel, G.: Translating HOL to Dedukti (2013), submitted
2. Bachmair, L., Dershowitz, N., Plaisted, D.: Completion without failure. In: Aït-Kaci, H., Nivat, M. (eds.) Resolution of Equations in Algebraic Structures, Volume 2: Rewriting Techniques. pp. 1–30. Academic Press inc. (1989)

3. Bachmair, L., Ganzinger, H.: Rewrite-based equational theorem proving with selection and simplification. Journal of Logic and Computation 4(3), 1–31 (1994)
4. Bachmair, L., Ganzinger, H.: Resolution theorem proving. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 19–99. Elsevier and MIT Press (2001)
5. Burel, G.: From axioms to rewriting rules, available on author's web page
6. Burel, G.: Experimenting with deduction modulo. In: Sofronie-Stokkermans, V., Bjørner, N. (eds.) CADE. LNCS, vol. 6803, pp. 162–176. Springer (2011)
7. Cousineau, D., Dowek, G.: Embedding pure type systems in the lambda-Pi-calculus modulo. In: Ronchi Della Rocca, S. (ed.) TLCA. LNCS, vol. 4583, pp. 102–117. Springer (2007)
8. Degtyarev, A., Voronkov, A.: Equality reasoning in sequent-based calculi. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 611–706. Elsevier and MIT Press (2001)
9. Delahaye, D., Doligez, D., Gilbert, F., Halmagrand, P., Hermant, O.: Zenon modulo: When Achilles outruns the tortoise using deduction modulo. In: McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.) LPAR. LNCS, vol. 8312, pp. 274–290. Springer (2013)
10. Dershowitz, N., Okada, M., Sivakumar, G.: Confluence of conditional rewrite systems. In: Jouannaud, J.P., Kaplan, S. (eds.) Proceedings 1st International Workshop on Conditional Term Rewriting Systems, Orsay (France). LNCS, vol. 308, pp. 31–44. Springer (July 1987)
11. Dowek, G.: What is a theory? In: Alt, H., Ferreira, A. (eds.) STACS. LNCS, vol. 2285, pp. 50–64. Springer (2002)
12. Dowek, G.: Confluence as a cut elimination property. In: Nieuwenhuis, R. (ed.) RTA. LNCS, vol. 2706, pp. 2–13. Springer (2003)
13. Dowek, G.: Polarized resolution modulo. In: Calude, C.S., Sassone, V. (eds.) IFIP TCS. IFIP AICT, vol. 323, pp. 182–196. Springer (2010)
14. Dowek, G., Hardin, T., Kirchner, C.: Theorem proving modulo. Journal of Automated Reasoning 31(1), 33–72 (2003)
15. Ganzinger, H., Stuber, J.: Superposition with equivalence reasoning and delayed clause normal form transformation. Inf. Comput. 199(1-2), 3–23 (2005)
16. Hermant, O.: Resolution is cut-free. Journal of Automated Reasoning 44(3), 245–276 (2009)
17. Hurd, J.: The OpenTheory standard theory library. In: Bobaru, M., Havelund, K., Holzmann, G.J., Joshi, R. (eds.) NFM 2011. LNCS, vol. 6617, pp. 177–191. Springer (2011)
18. Jacquel, M., Berkani, K., Delahaye, D., Dubois, C.: Tableaux modulo theories using superdeduction – an application to the verification of B proof rules with the Zenon automated theorem prover. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR. LNCS, vol. 7364, pp. 332–338. Springer (2012)
19. Knuth, D.E., Bendix, P.B.: Simple word problems in universal algebras. In: Leech, J. (ed.) Computational Problems in Abstract Algebra, pp. 263–297. Pergamon Press, Oxford (1970)
20. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the ACM 12, 23–41 (1965)
21. Vorobyov, S.G.: On the arithmetic inexpressiveness of term rewriting systems. In: LICS. pp. 212–217 (1988)