CENTRE de RECHERCHE en INFORMATIQUE de NANCY

Campus Scientifique
Bâtiment LORIA
Boîte Postale 239
54506 Vandœuvre-lès-Nancy Cedex

Téléphone : (+ 33) 83.59.20.00
Télécopie : (+ 33) 83.41.30.79

Automated Deduction with Associative Commutative Operators

Michaël Rusinowitch and Laurent Vigneron

CRIN 93-R-252

NOVEMBER 1993

Published in the Journal of
Applicable Algebra in Engineering, Communication and Computation [RV95]

Automated Deduction with Associative Commutative Operators [1]

Michaël Rusinowitch and Laurent Vigneron [2]
CRIN & INRIA-Lorraine
BP239, 54506 Vandœuvre-lès-Nancy Cedex, France
e-mail : {Michael.Rusinowitch, Laurent.Vigneron}@loria.fr

Abstract : We propose a new inference system for automated deduction with equality and associative commutative operators. This system is an extension of the ordered paramodulation strategy. However, rather than using associativity and commutativity as the other axioms, they are handled by the AC-unification algorithm and the inference rules. Moreover, we prove the refutational completeness of this system without needing the functional reflexive axioms or AC-axioms. Such a result is obtained by semantic tree techniques. We also show that the inference system is compatible with simplification rules.

Keywords : Automated Deduction, Associative and Commutative Theories, Paramodulation, Resolution, Term Rewriting

Déduction Automatique avec des Opérateurs Associatifs et Commutatifs

Résumé : Nous proposons un nouveau système d'inférence pour la déduction automatique avec le prédicat d'égalité et des opérateurs associatifs et commutatifs. Ce système est une extension de la stratégie de paramodulation ordonnée. L'associativité et la commutativité ne figurent pas comme les autres axiomes. Elles sont gérées par l'algorithme d'AC-unification et les règles d'inférence. Nous prouvons la complétude réfutationnelle de ce système par des techniques d'arbres sémantiques. Nous montrons également que notre système est compatible avec des règles de simplification.

Mots clés : Déduction automatique, Théories associatives et commutatives, Paramodulation, Résolution, Réécriture

[1] A preliminary version of this paper was presented in the International Workshop FAIR'91 [RV91]

[2] This work is partially supported by the GDR Programmation and the Esprit Basic Research working group 6028, CCL.

Contents

1 Introduction

Automated deduction with equality and associative commutative (AC) operators (i.e. binary operators f satisfying the axioms : $f(f(x,y),z) \simeq f(x,f(y,z))$ and $f(x,y) \simeq f(y,x)$) has been considered as a difficult problem. The reason is that the presence of AC-axioms increases dramatically the number of possible deductions. For instance, there are 1680 ways to write the term $f(t_1, f(t_2, f(t_3, f(t_4, t_5))))$, where f is an AC-operator and t_1, t_2, t_3, t_4, t_5 are different constants.

The approach we propose for dealing with AC-axioms is to work in the AC-congruence classes, to employ associative commutative identity checking, pattern matching and unification, and to work only at useful positions (for AC-operators), notion introduced by Lai [Lai89] and applied to completion modulo AC by Domenjoud [Dom91]. This idea of building axioms within the unification procedures has been first initiated by Plotkin [Plo72]. In the context of automated deduction, it has been investigated too by Lankford [Lan79a], Stickel [Sti84], Anantharaman and al. [AHM89], Petermann [Pet91]. However these works essentially refer to practical experimentations, and do not account for completeness results of the inference systems that they study. In the following, we focus on giving a complete set of inference rules for first order logic with equality and built-in AC-unification. The recent development of efficient AC-unification algorithms [Sti81, BHK+88] strongly argues in favour of the effectiveness of our method.

Our inference system includes resolution and paramodulation to deal with equality. Paramodulation performs substitution directly by replacing one argument of an equality atom by the other one, when the former occurs in some clause ; it has been introduced by [RW69]. Some important refinements have been proposed by introducing a simplification ordering on terms and forbidding the replacement of a term by a bigger one. These aspects are fully developed in [Pet83, HR91]. Here, we also confine the term replacement step of our paramodulation rules by a simplification ordering. However, for the sake of completeness we also require this ordering to be total on the set of AC-congruence classes and to be AC-compatible.

The refutational completeness of our set of rules is derived by the transfinite semantic tree method of [Rus89]. Semantic trees represent the set of Herbrand interpretations for formulas in clausal form. When a tree associated to a formula is empty then we can be sure that the empty clause can be derived, and therefore the initial formula is unsatisfiable. We have been aware recently that a similar result has been derived independently by Paul [Pau92]. His proof, unlike ours, involves some reasoning with AC-congruence classes, and relies on an unpublished work of Lankford (*Canonical inference, 1975*). Also, Paul has not proved the compatibility of his system with simplification rules. More recently, Wertz [Wer92] has developed two methods for theorem proving modulo a set of equations E. These methods, unlike ours, create explicitly extended clauses from an initial set of clauses. We do not know yet how these methods compare with ours.

Associative commutative theories have been thoroughly studied in the context of term rewriting systems. We will not review here the Knuth-Bendix method [KB70]. Let us just mention that it has been extended to incorporate associativity and commutativity by [LB77, PS81, JK86].

The layout of this paper is as follows : Section 2 presents an overview of our approach on two examples. Section 3 summarizes the basic material which is relevant to this work. In particular we give some details on the construction of orderings which are AC-compatible and total on the Herbrand Universe. These orderings are fundamental to build transfinite E-semantic trees (Section 4), and to define the refutationally complete inference system described in Section 5. The proof of refutational completeness is developed in Section 6. Moreover, we introduce reduction rules, like subsumption and simplification, and we prove that they maintain the refutational completeness in Section 7. Section 8 gives some information and an example of resolution of `DATAC`, system that is an implementation of our inference rules. The software is written in CAML Light and runs on SUN, HP and IBM PC Workstations.

2 Introducing Examples

In this section, we first show a simple example to introduce associative and commutative theories and to explain the main ideas of our method. Then, we can see that non-trivial consequences can be hidden by the AC-axioms.

2.1 A Simple Example

Here is a simple example to show the problems due to the presence of the AC-axioms and to present our approach for solving them. We consider the following system of equations S :

$$\begin{cases} a+b \simeq d & \textit{(1)} \\ c+b \simeq e & \textit{(2)} \\ g(c+d) \simeq h(a) & \textit{(3)} \\ g(e+a) \simeq h(b) & \textit{(4)} \end{cases}$$

assuming that $+$ is an AC-operator, a, b, c, d and e are constants, and g and h are unary operators. Now, let us prove the following theorem : $h(a) \simeq h(b)$.
The first step is to add to S the inequation $h(a) \neq h(b)$ *(Th)* and let us try to find a contradiction.

With the classical paramodulation method, we just add the $AC(+)$ axioms to S and perform inferences in the empty theory, i.e. with syntactic unification.

$$\begin{cases} (x+y)+z \simeq x+(y+z) & \textit{(A)} \\ x+y \simeq y+x & \textit{(C)} \end{cases}$$

The refutation of the new system is performed as follows :

para (2, A)	$e+z \simeq c+(b+z)$	*(5)*
para (C, 5)	$e+z \simeq c+(z+b)$	*(6)*
para (1, 6)	$e+a \simeq c+d$	*(7)*
para (7, 4)	$g(c+d) \simeq h(b)$	*(8)*
para (4, 8)	$h(a) \simeq h(b)$	*(9)*
resol (9, Th)	$\square$	*(10)*

where *para (i, j)* means a paramodulation from the clause *(i)* into the clause *(j)*, and *resol (i, j)* means a resolution between the clauses *(i)* and *(j)*.

We notice that a third of the steps needed for the refutation of the system uses the AC-axioms. However, when dropping these axioms we must replace them by other mechanisms. For instance, using unification modulo AC allows to suppress the step using commutativity. However modifying the unification algorithm is not sufficient. The last problem is to avoid the creation of extended equations, which results from a paramodulation in the axiom of associativity. Extended equations (or clauses) dramatically increase the number of possible deductions. Therefore, instead of introducing a special control (as the *protection* of extended rules in AC-completion procedures [PS81, JK86]), we rather build these extended equations on the fly, that is, when they are immediately followed by a paramodulation step. Hence, we have designed an inference rule, named *extended paramodulation*, which, given two clauses $(f(A,B)\simeq C)\vee P$ and $(f(D,E)\simeq F)\vee Q$, generates an instance of the clause $(f(C,G)\simeq f(F,H))\vee P\vee Q$ whenever f is an AC-operator and $f(f(A,B),G)$ and $f(f(D,E),H)$ are unifiable modulo AC. This rule may be viewed as a generalization of the superposition rule of the Buchberger algorithm for computing Gröbner bases. Relations between Gröbner bases technique and term rewriting techniques have been studied in [Bün91].

Now the refutation of the previous system is carried out as follows :

ext-para (2, 1)	$a+e\simeq d+c$	*(5)*
para (5, 3)	$g(a+e)\simeq h(a)$	*(6)*
para (4, 6)	$h(b)\simeq h(a)$	*(7)*
resol (7, Th)	$\square$	*(8)*

Hence we have reached our goal : all the deduction steps involving AC-axioms have now disappeared. This very last refutation is the one that can be obtained when applying our inference rules to be introduced in Section 5.

2.2 A Non-Trivial Example

Example 2.1 has introduced mechanisms for simulating AC reasoning. Here, we apply these mechanisms to a system which is more difficult to solve. Let S be the set of clauses :

$$\begin{cases} x+a\simeq b & \textit{(1)} \\ P(b+c) & \textit{(2)} \\ \neg P(b) & \textit{(3)} \end{cases}$$

where a, b and c are constants, $+$ is an AC-operator, and P is a predicate symbol.
We assert that S is incoherent in AC-theories and we prove it as follows :

ext-para (1, 1)	$x+b\simeq b+y$	*(4)*
para (1, 4)	$x+b\simeq b$	*(5)*
para (5, 2)	$P(b)$	*(6)*
resol (6, 3)	$\square$	*(7)*

Let us detail the extended paramodulation step :

$$\begin{array}{rcl} x+a & \simeq & b \\ & \downarrow & \text{extension of } \textit{(1)} \\ x+a+y & \simeq & b+y \\ & \downarrow & \text{reduction by } \textit{(1)} \\ x+b & \simeq & b+y \end{array}$$

More generally, the notion of extended equation (or clause) is essential in AC-theories, in order to reveal some consequences that follow from the AC-axioms.

3 Terms and Orderings

3.1 Notations and Preliminary Notions

Let $\mathcal{F}$ be a finite set of functions with arities, and let $\mathcal{X}$ be a countably infinite set of variables. The algebra of terms composed from $\mathcal{F}$ and $\mathcal{X}$ is denoted by $\mathcal{T}(\mathcal{F},\mathcal{X})$. We use $\mathcal{T}(\mathcal{F})$ for the set of ground terms (the *Herbrand universe*).

Let $\mathcal{P}$ be a finite set of predicate symbols including the equality predicate "$\simeq$". The set of *atoms* $\mathcal{A}(\mathcal{P},\mathcal{F},\mathcal{X})$ is $\{p(t_1,\ldots,t_n) \mid p \in \mathcal{P} \textit{ and } t_i \in \mathcal{T}(\mathcal{F},\mathcal{X})\}$. We denote the set of ground atoms (the *Herbrand base*) by $\mathcal{A}(\mathcal{P},\mathcal{F})$. An *equality atom* is an atom whose predicate symbol is "$\simeq$". Throughout this paper, we assume that "$\simeq$" is commutative in the sense that we do not distinguish between the atoms $(s \simeq t)$ and $(t \simeq s)$. However, usually, we write $(s \simeq t)$, instead of $(t \simeq s)$, when s is greater than t for some ordering on terms. A *literal* is either an atom (A) or the negation of an atom $(\neg A)$, and a *clause* is a multiset of literals. In general we use the term *object* to indicate a term, an atom, a literal, or a clause, and the term *ground object* to indicate a ground term, a ground atom, a ground literal, or a ground clause. For a clause C, *Atoms(C)* represents the multiset of its atoms.

We assume that the operators from a given subset $\mathcal{F}_{AC}$ of $\mathcal{F}$ are associative and commutative, which means that for $f \in \mathcal{F}_{AC}$ the following axioms are implicit in the theory to be considered :

$$AC(f) \quad \left\{ \begin{array}{ccc} f(f(x,y),z) & \simeq & f(x,f(y,z)) \\ f(x,y) & \simeq & f(y,x) \end{array} \right.$$

The congruence on $\mathcal{T}(\mathcal{F},\mathcal{X})$, generated by the associative commutative equations satisfied by the symbols in $\mathcal{F}_{AC}$, is written $=_{AC}$ and is called AC-equality. The syntactic equality of objects is denoted $=$.

Let $\mathcal{V}ar(t)$ denote the set of variables appearing in an object t. A *substitution* is a mapping σ from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\sigma(x) \neq x$ for only finitely many variables. We use $\mathcal{D}om(\sigma)$ to denote the set $\{x \mid \sigma(x) \neq x\}$. We further assume that for every $x \in \mathcal{D}om(\sigma)$, $\mathcal{V}ar(\sigma(x)) \cap \mathcal{D}om(\sigma) = \emptyset$. The substitution σ is *applied* to an object t if all variables x in t are replaced by $\sigma(x)$. The result is denoted by $t\sigma$. A substitution σ is an *AC-unifier* of two objects s and t if $s\sigma =_{AC} t\sigma$. The set of *most general AC-unifiers* (*AC-mgus*) of two terms s and t is defined by : for every AC-unifier θ of s and t, there exists an AC-mgu ρ such that θ is AC-equal to the composition of σ and ρ (written $\rho\sigma$) ; it means that : for each term u, $u\theta =_{AC} (u\rho)\sigma$. In the empty theory, a mgu is unique upon renaming of variables.

To express subterms and substitutions more effectively, we use *positions*. Envision a term represented as a tree ; a *position* (or *occurrence*) in a term indicates a node in the tree. Positions are usually represented as sequences of integers. Let p be a position, we use $t_{|_p}$ for the subterm of t at p. More precisely, $t_{|_\varepsilon} = t$ where ε is the empty position, and $g(t_1,\ldots,t_n)_{|_{i.p}} = {t_i}_{|_p}$. We also use $s = s[p \leftarrow t]$ (or $s[t]_p$) to denote that s is a term whose subterm at position p is t. For convenience, we express it by $s = s[t]$ if the particular position is not important. A subterm of t is *proper* if it is distinct from t. The function $\mathcal{H}ead(t)$ returns the functional operator at the head of the term t, if t is not a variable ; otherwise, $\mathcal{H}ead(x)$ is undefined.
$\mathcal{MP}os(t)$ denotes the set of all non variable positions p in t such that either $\mathcal{H}ead(t_{|_p}) \notin \mathcal{F}_{AC}$, or, if $p = p'.i$, $\mathcal{H}ead(t_{|_p})$ and $\mathcal{H}ead(t_{|_{p'}})$ are not the same AC-operator. This corresponds to the notion of useful positions defined by Lai [Lai89] and Domenjoud [Dom91].

We denote by $\subseteq$ (resp. $\cap_|$ and resp. $\cup_|$) the relation of inclusion (resp. intersection and resp. union) of multisets, and by $\subseteq_{AC}$, $\cap_{|AC}$ and $\cup_{|AC}$ the analogous relations where AC-equality replaces syntactic equality for comparing objects.

We recursively define $\mathcal{H}terms(t, f)$, where t is a term and f an AC-operator, by : if t is a variable or $\mathcal{H}ead(t) \neq f$, then $\mathcal{H}terms(t, f)$ is equal to $\{t\}$; else, $\mathcal{H}terms(t, f)$ is the multiset $\mathcal{H}terms(t_{|_1}, f) \cup_{|AC} \mathcal{H}terms(t_{|_2}, f)$, since an AC-operator has exactly two arguments. For instance, $\mathcal{H}terms(f(a, f(f(a, x), g(b))), f) = \{a, a, x, g(b)\}$.

Given four terms l_1, l_2, w_1 and w_2 such that $w_1 =_{AC} w_2$, $w_1 = w_1[l_1]$ and $w_2 = w_2[l_2]$, we say that l_1 and l_2 are at *AC-dependent positions* if there is no term w, AC-equal to w_1, such that : $w = w[l_1]_{p_1} = w[l_2]_{p_2}$, and p_1 and p_2 are independant positions, i.e. $p_1 \neq q.p_2$ and $p_2 \neq q.p_1$, for a position q. Such a term w will be written $w[l_1]_{p_1}[l_2]_{p_2}$.

3.2 Compatible Orderings for Associative-Commutative Theories

Orderings are used to define restricted versions of resolution and paramodulation. Firstly, resolution and paramodulation need only to be performed on the maximal literals. Secondly, when using an equality in a clause to paramodulate, only the largest of the two terms of this equality needs to be considered. Our set of inference rules, to be introduced in the next section, can be proved refutationally complete if it is defined with respect to a *complete simplification ordering*.

Let us recall the definition of these orderings :

Definition 1 *A transitive irreflexive relation $>$ on the set of terms is a complete simplification ordering (CSO, for short) if*

1. *$>$ is total on the set of ground terms*
2. *$>$ is well-founded*
3. *(monotonicity) $s > t$ implies $w[s] > w[t]$*
4. *(subterm) for any proper subterm s of t, we have $s < t$*
5. *(stability) for any substitution σ, $s > t$ implies $s\sigma > t\sigma$*

Now, since we want to perform inferences on literals representing AC-congruence classes, these inferences should be somewhat independent of the chosen congruence class representatives. That is why we also require our ordering to have the *AC-compatibility* property :

Definition 2 *An ordering $>$ on $\mathcal{T}(\mathcal{F})$ is AC-compatible if whenever we have $s > t$ and $s \neq_{AC} t$, we also have $s' > t'$, for any ground terms s' and t' respectively AC-equal to s and t.*

The design of AC-compatible orderings for proving termination of rewrite systems modulo AC has been considered as a hard task. In fact, to our knowledge, very few constructions are available in the literature. Perhaps the best known among them is the *associative path ordering* scheme [BP85], which extends the *recursive path ordering* (see also [Der82]). However, this ordering puts serious limitations on the precedence of AC-symbols. In fact two AC-symbols cannot be compared in the precedence unless they are related by a distributivity law. This explains why it seems difficult to extend the *associative path ordering* to get a total ordering when there are several AC-symbols.

Up to now only one AC-compatible complete simplification ordering for a signature which contains any number of AC-symbols has been found. It is described in [NR91] and is based on the method of polynomial interpretations [Lan79b, BCL87]. This ordering could be used for our purpose. In particular, inference rules built on it are refutationally complete.

In the case of **one** single AC-operator, many more constructions of total AC-compatible orderings are available.

Given an AC-compatible CSO on terms, we define :

Definition 3 *Let $>$ be an AC-compatible CSO on terms. We define another AC-compatible ordering $\succ$ by :* $s \succ t$ *iff* $s > t$ *and* $s \neq_{AC} t$.

Then, we extend the ordering $>$ to atoms by the following definition, that preserves the property of AC-compatibility on atoms.

Definition 4 *Let $>_p$ be a total precedence on operators, "$\simeq$" being the smallest predicate. Let $>$ be an AC-compatible CSO on terms. The ordering $>_A$ is defined on the set of atoms by :*

$P(s_1, \ldots, s_n) >_A Q(t_1, \ldots, t_m)$ *if*

- *either* $P >_p Q$,
- *or* $P = Q$, P *is not the equality predicate, and*
  . *either* $(s_1, \ldots, s_n) \succ^{lex} (t_1, \ldots, t_n)$,
    where $\succ^{lex}$ *is the lexicographic extension of* $\succ$,
  . *or* $P(s_1, \ldots, s_n) =_{AC} Q(t_1, \ldots, t_n)$, *and* $(s_1, \ldots, s_n) >^{lex} (t_1, \ldots, t_n)$,
    where $>^{lex}$ *is the lexicographic extension of* $>$,
- *or* $P = Q$, P *is the equality predicate, and, assuming that* $s_1 \geq s_2$ *and* $t_1 \geq t_2$,
  . *either* $s_1 \succ t_1$,
  . *or* $s_1 =_{AC} t_1$, *and* $s_2 \succ t_2$,
  . *or* $s_1 =_{AC} t_1$, $s_2 =_{AC} t_2$, *and* $\{s_1, s_2\} \gg \{t_1, t_2\}$,
    where $\gg$ *is the multiset extension of* $>$.

It is easy to check that this ordering on atoms is a CSO. As for $>$, we define another ordering on atoms by :

Definition 5 *Let $>_A$ be a complete simplification ordering on atoms as defined above. The ordering $\succ_A$ is defined by :* $A \succ_A B$ *if* $A >_A B$ *and* $A \neq_{AC} B$.

We shall compare literals by forgetting signs and comparing atoms. For simplicity, the ordering on literals will be denoted $\succ_A$ too.

4 Transfinite Semantic Trees

The problem of proving the completeness of theorem proving strategies involving equality has been prominent in automated theorem proving since its first conception. A notorious instance is the question of whether the inference system consisting of resolution and paramodulation is complete without the functionally reflexive axioms and without paramodulating into variables. In Brand [Bra75] an indirect proof of uselessness of functional reflexive axioms was described (as a corollary of the completeness of the modification method). A direct proof by semantic trees was given in [Pet83]. However, Peterson's proof requires the use of a simplification ordering which is also order isomorphic to ω on ground atoms. Hsiang and Rusinowitch [HR91] have developed a new method based on transfinite semantic trees for relaxing this condition and permitting a larger class of orderings.

This method has also been applied to Knuth-Bendix completion procedure [HR87] and to conditional completion [KR87].

This is the method that we shall use here for proving the refutational completeness of the inference rules which will be introduced in the next section. This section introduces some main notions on E-interpretations and transfinite semantic trees ; for more details, see [HR91]. Let us just mention that some other techniques have been proposed by Pais and Peterson [PP91] and Bachmair and Ganzinger [BG90]. The first one is based on forcing and, as ours, it uses transfinite induction to build the Herbrand model for a set of clauses. The second one is based on the use of canonical rewrite systems to represent equality interpretations.

The Herbrand base $\mathcal{A}(\mathcal{P},\mathcal{F})$ can be ordered as an increasing sequence $\{A_i\}_{i<\lambda}$ by $>_A$ (λ being its ordinal). Given an atom A_α, we write W_α the initial segment $\{A_i \mid i < \alpha\}$. The successor of the ordinal α is denoted by $\alpha + 1$.

Given the ordinal α, a *partial E-interpretation* on W_α is a mapping I (sometimes written I_α) from W_α to $\{T, F\}$ satisfying :

- if $(s \simeq s) \in W_\alpha$, then $I(s \simeq s) = T$
- if $(s \simeq t)$, $B[s]$, $B[t] \in W_\alpha$ and $I(s \simeq t) = T$, then $I(B[s]) = I(B[t])$

An *E-interpretation* is a partial E-interpretation defined on W_λ, the entire Herbrand base. We extend E-interpretations to the set of ground clauses in the usual way as follows : let I be a (partial) E-interpretation on W_α, A an element in W_α, and $C = L_1 \vee \ldots \vee L_n$ a ground clause whose atoms are all in W_α. Then, $I(\neg A) = \neg I(A)$ and $I(C) = I(L_1) \vee \ldots \vee I(L_n)$.

Given an E-interpretation I and a clause C, I *E-satisfies* C (or C is *valid* in I) if for every ground instance C' of C, $I(C') = T$. Otherwise, we say that I *falsifies* C. C is *E-satisfiable* if C is valid in some E-interpretation. Otherwise it is *E-unsatisfiable*.

Given a set of clauses S, S is *E-satisfiable* if there is an E-interpretation I such that I satisfies every clause in S. Otherwise S is *E-unsatisfiable*.

In associative and commutative theories, a set of clauses S is *AC-unsatisfiable* if $S \cup AC$ is E-unsatisfiable, where AC is the set of associativity and commutativity axioms of the operators of $\mathcal{F}_{AC}$.

Let v and w be two ground atoms and let I be an E-interpretation on W_α. We say that w is *I-reducible* to v by $(s \simeq t)$, and we write it $w \longrightarrow_I^{s \simeq t} v$, if there is an atom $(s \simeq t) \in W_\alpha$ such that :

$$w = w[s] \ , \ s > t \ , \ w >_A (s \simeq t) \ , \ I(s \simeq t) = T \ \text{ and } \ v = w[t]$$

An atom which is not I-reducible is said I*-irreducible*. The following theorem states that to test I-reducibility, it is sufficient to consider I-irreducible equalities.

Theorem 6 (Reduction Theorem) *A ground atom w is I-reducible if and only if it is I-reducible by an I-irreducible equality.*

By the next theorem, it is possible to build inductively all the E-interpretations in a manner which is similar to that in [Pet83].

Theorem 7 [Pet83] *Let $I : W_{\alpha+1} \to \{T, F\}$ be such that I is an E-interpretation on W_α. Let J be the restriction of I to W_α. Then, I is an E-interpretation on $W_{\alpha+1}$ iff :*

(1) *A_α is J-reducible to an atom B and $I(A_\alpha) = I(B)$, or*

(2) *A_α is J-irreducible, of the form $(t \simeq t)$, and $I(A_\alpha) = T$, or*

(3) *A_α is J-irreducible and not of the form $(t \simeq t)$.*

Let I and J be two E-interpretations defined as in the previous theorem. We say that I *is an extension of* J. The collection of all partial E-interpretations is called a *transfinite E-semantic tree*. A *node* is an element of the tree.

Let T be a transfinite E-semantic tree. We call *maximal consistent E-semantic tree* of a set of clauses S, and we write *MCT(S)*, the maximal subtree of T such that :

> For every node I in *MCT(S)*, every clause C in S, and every clause C', AC-equal to a ground instance of C and whose atoms are in the domain of I, $I(C') = T$.

A *path* is a sequence of nodes $(I_i)_{i \leq \alpha}$ such that α is an ordinal ($\leq \lambda$) and the domain of every I_i is W_i. A *failure node* is a node which falsifies a clause C. In particular, if J is the last node of a path of *MCT(S)* then every extension of J is a failure node. A *maximal path* in *MCT(S)* is a path whose extensions are not in *MCT(S)* (hence these extensions are failure nodes).
The following lemma states that a failure node cannot be the limit of non failure nodes. Its proof is similar to the classical one (see [HR91]).

Lemma 8 (Closure Lemma) *Let S be a set of clauses. Then, every maximal path of MCT(S) has a last element (in MCT(S)).*

A consequence of this lemma is :

Corollary 9 *Let S be a set of clauses ; then, S is AC-unsatisfiable if and only if every maximal path in $MCT(S \cup AC)$ extends to a failure node, where AC is the set of all AC-axioms.*

Adding AC to S is necessary in order to introduce failure nodes when an interpretation falsifies an equation $(u \simeq u')$ where $u =_{AC} u'$. Such an interpretation may be consistent but not AC-consistent and it should be discarded from the set of potential models of S.

Let INF be a set of inference rules and S a set of clauses. Let *INF(S)* denote the set of clauses obtained by adding to S all clauses generated by applying some rule of INF to S. Let $INF^0(S) = S$, $INF^{n+1}(S) = INF(INF^n(S))$, and $INF^*(S) = \bigcup_{n \geq 0} INF^n(S)$.

A set of inference rules INF is *refutationally complete for the AC theories*, or *AC-complete* for short, if, given any AC-unsatisfiable set of clauses S, $INF^*(S)$ contains the empty clause.

Here is the theorem of completeness of inference rules for E-semantic trees in AC theories :

Theorem 10 (Fundamental Theorem) *A set of inference rules INF is AC-complete if and only if $MCT(INF^*(S) \cup AC)$ contains only the empty interpretation whenever S is AC-unsatisfiable.*

We can remark that if $MCT(INF^*(S) \cup AC)$ is empty (contains only the empty interpretation), i.e. $INF^*(S)$ contains the empty clause, then $MCT(INF^*(S))$ is empty too.

5 Inference Rules and Lifting Lemma

The inference rules, that we shall define in this section, are compatible with the strategy of ordered clauses presented in [Pet83, Rus89]. This strategy permits application of the paramodulation inference rule under a refined form : a term cannot be replaced by a more complex one, along a paramodulation step ; in particular, we never paramodulate into a variable. After the definitions of inferences rules, we shall prove the lifting lemma.

From now on, we assume that we are given an AC-compatible CSO on terms $>$ and its extension to atoms $>_A$, as in definition 4. In the following inference rules, orderings refer to definitions 3 and 5. For instance, the notation $s \preceq t$ means that either $s \prec t$, or $s =_{AC} t$. And $s \not\preceq t$ means that either $s \succ t$ or s and t are incomparable ; on ground objects, by totality, $s \not\preceq t$ is equivalent to $s \succ t$.

5.1 Inference Rules

Let us define our inference rules. Note that they require AC-unification, and that f is an AC-operator in the examples which illustrate them.

Definition 11 (AC-Factoring)

$$\frac{L_1 \vee \ldots \vee L_n \vee D}{(L_1 \vee D)\sigma} \qquad \textit{if} \quad \begin{cases} \sigma \in AC\_mgus\{L_1, \ldots, L_n\} \\ \forall A \in Atoms(D),\ L_1\sigma \not\preceq_A A\sigma \end{cases}$$

Comments : *this ordered AC-factoring rule is applied if σ is a most general AC-unifier of the n literals $L_1, \ldots, L_n$, and if there does not exist an atom A of D such that $A\sigma$ is greater than or equal to the atom corresponding to $L_1\sigma$, for the ordering $\succ_A$.*

Example :

$$\frac{P(f(a,y)) \vee P(f(b,x)) \vee Q(x,y)}{P(f(a,b)) \vee Q(a,b)} \qquad \textit{where} \quad \sigma = \{y \leftarrow b, x \leftarrow a\}$$

Definition 12 (AC-Reflexion)

$$\frac{\neg(s \simeq t) \vee D}{D\sigma} \qquad \textit{if} \quad \begin{cases} \sigma \in AC\_mgus\{s, t\} \\ \forall A \in Atoms(D),\ (s \simeq t)\sigma \not\preceq_A A\sigma \end{cases}$$

Comments : *this ordered AC-reflexion rule is applied if σ is a most general AC-unifier of s and t, and if there does not exist an atom A of D such that $A\sigma$ is greater than or equal to $(s \simeq t)\sigma$, for the ordering $\succ_A$.*

Example :

$$\frac{\neg(f(a,y) \simeq f(b,x)) \vee (f(x,y) \simeq c)}{(f(a,b) \simeq c)} \qquad \textit{where} \quad \sigma = \{y \leftarrow b, x \leftarrow a\}$$

Definition 13 (AC-Resolution)

$$\frac{L_1 \vee D_1 \qquad \neg L_2 \vee D_2}{(D_1 \vee D_2)\sigma} \qquad \textit{if} \quad \begin{cases} \sigma \in AC\_mgus\{L_1, L_2\} \\ \forall A \in Atoms(D_1),\ L_1\sigma \not\preceq_A A\sigma \\ \forall A \in Atoms(D_2),\ L_2\sigma \not\preceq_A A\sigma \end{cases}$$

Comments : *this ordered AC-resolution rule is applied if σ is a most general AC-unifier of L_1 and L_2, and if $L_1\sigma$, respectively $L_2\sigma$, is not less than or equal to an atom of $D_1\sigma$, resp. $D_2\sigma$.*

Example :

$$\frac{P(f(a,y)) \vee Q(y) \qquad \neg P(f(b,x)) \vee Q(x)}{Q(b) \vee Q(a)} \qquad \textit{where} \quad \sigma = \{y \leftarrow b, x \leftarrow a\}$$

Definition 14 (AC-Paramodulation)

$$\frac{(s \simeq t) \vee D_1 \qquad L \vee D_2}{(L[t]_p \vee D_1 \vee D_2)\sigma} \qquad \text{if} \begin{cases} \sigma \in AC\_mgus\{L_{|_p}, s\} \quad \textit{where } p \in \mathcal{MP}os(L) \\ \forall A \in Atoms(D_1),\ (s \simeq t)\sigma \not\preceq_A A\sigma \\ \forall A \in Atoms(D_2),\ L\sigma \not\preceq_A A\sigma \\ s\sigma \not\preceq t\sigma \end{cases}$$

Comments : *this ordered AC-paramodulation rule applies if there is a non variable occurrence p of the literal L and a most general AC-unifier σ of $L_{|_p}$ and s. Moreover, $L\sigma$, resp. $(s \simeq t)\sigma$, is not less than or equal to an atom of $D_2\sigma$, resp. $D_1\sigma$. Another condition is that $t\sigma$ is not greater than or equal to $s\sigma$. The position p has to be a maximal occurrence in L of the head operator of $s\sigma$, if it is AC.*

Example :

$$\frac{(f(a, y) \simeq c) \vee (g(y) \simeq d) \qquad P(f(b, x)) \vee Q(x)}{P(c) \vee (g(b) \simeq d) \vee Q(a)} \qquad \textit{where} \quad \sigma = \{y \leftarrow b, x \leftarrow a\}$$

Definition 15 (AC-Contextual Paramodulation)

$$\frac{(s \simeq t) \vee D_1 \qquad L \vee D_2}{(L[f(t, x)]_p \vee D_1 \vee D_2)\sigma}$$

$$\text{if} \begin{cases} \sigma \in AC\_mgus\{L_{|_p}, f(s, x)\} \\ \qquad \textit{where } f = \mathcal{H}ead(s\sigma) \in \mathcal{F}_{AC}, \\ \qquad\qquad x \textit{ is a new variable, } p \in \mathcal{MP}os(L) \\ \forall A \in Atoms(D_1),\ (s \simeq t)\sigma \not\preceq_A A\sigma \\ \forall A \in Atoms(D_2),\ L\sigma \not\preceq_A A\sigma \\ s\sigma \not\preceq t\sigma \\ \forall w \in \mathcal{H}terms(L_{|_p}, f),\ \mathcal{H}terms(s\sigma, f) \not\subseteq_{AC} \mathcal{H}terms(w\sigma, f) \end{cases}$$

Comments : *this ordered AC-contextual paramodulation rule applies if there is a position p of the literal L and a most general AC-unifier σ of $L_{|_p}$ and $f(s, x)$, where f is the AC-operator at the head of $s\sigma$; p has to be a maximal occurrence of f in L (also p is not a variable position). Moreover, $L\sigma$, resp. $(s \simeq t)\sigma$, is not less than or equal to an atom of $D_2\sigma$, resp. $D_1\sigma$. Another condition is that $t\sigma$ is not greater than or equal to $s\sigma$. The last condition implies that the term $s\sigma$ was not introduced by the substitution in a subterm of L.*

Extending the term s and applying AC-unification allows us to detect when some equality replacement is possible, whichever representatives (modulo AC) have been chosen for the clauses.

Example :

$$\frac{(f(a, b) \simeq c) \vee (g(b) \simeq d) \qquad P(f(b, f(y, d))) \vee Q(y)}{P(f(c, d)) \vee (g(b) \simeq d) \vee Q(a)} \qquad \textit{where} \quad \sigma = \{y \leftarrow a, x \leftarrow d\}$$

Definition 16 (AC-Extended Paramodulation)

$$\frac{(s \simeq t) \vee D_1 \qquad (l \simeq r) \vee D_2}{((f(t, x) \simeq f(r, y)) \vee D_1 \vee D_2)\sigma}$$

$$
\text{if} \left\{ \begin{array}{l}
\sigma \in AC\_mgus\{f(s,x), f(l,y)\} \\
\qquad \textit{where } \mathcal{H}ead(s\sigma) = \mathcal{H}ead(l\sigma) = f \in \mathcal{F}_{AC} \;, \\
\qquad\qquad x \textit{ and } y \textit{ are new variables} \\
\forall A \in Atoms(D_1),\ (s \simeq t)\sigma \not\succeq_A A\sigma \\
\forall A \in Atoms(D_2),\ (l \simeq r)\sigma \not\succeq_A A\sigma \\
s\sigma \not\preceq t\sigma \quad \textit{and} \quad l\sigma \not\preceq r\sigma \\
\mathcal{H}terms(s\sigma, f) \cap_{|AC} \mathcal{H}terms(l\sigma, f) \;\neq\; \emptyset \\
\mathcal{H}terms(x\sigma, f) \cap_{|AC} \mathcal{H}terms(y\sigma, f) \;=\; \emptyset
\end{array} \right.
$$

Comments : *this ordered AC-extended paramodulation rule can be seen as a contextual paramodulation from* $(s \simeq t) \vee D_1$ *into* $(f(l,y) \simeq f(r,y)) \vee D_2$ *, at the head of* $f(l,y)$*, with* σ *as a most general AC-unifier. Moreover,* $r\sigma$ *is not greater than or equal to* $l\sigma$*.*
The last two conditions force the overlap between $s\sigma$ *and* $l\sigma$ *to be a non-trivial one. For instance, the first one means that* $s\sigma$ *and* $l\sigma$ *must share a maximal subterm. Sufficiency of these two restrictions is shown in the construction of the quasi-rightmost path in the proof of lemma 30 (Section 6.2).*

Example :

$$
\frac{(f(a,b) \simeq c) \vee (g(b) \simeq d) \qquad (f(a,d) \simeq e) \vee (h(d) \simeq b)}{(f(c,d) \simeq f(e,b)) \vee (g(b) \simeq d) \vee (h(d) \simeq b)} \qquad \textit{where} \quad \sigma = \{x \leftarrow d, y \leftarrow b\}
$$

We emphasize the role of conditions on $\mathcal{H}terms$ by two examples. They show that, when these conditions are not satisfied, the generated clause is redundant. In other words, this clause is not necessary for deriving a contradiction from an AC-unsatisfiable set of clauses.

1. *If* $s\sigma$ *and* $l\sigma$ *have no common* $\mathcal{H}terms$ *:*

$$
\frac{(f(a,b) \simeq c) \qquad (f(d,e) \simeq g)}{(f(c, f(d,e)) \simeq f(g, f(a,b)))} \qquad \textit{where} \quad \sigma = \{x \leftarrow f(d,e), y \leftarrow f(a,b)\}
$$

$$
\textit{Here,} \qquad \left\{ \begin{array}{l}
\mathcal{H}terms(s\sigma, f) = \{a, b\} \\
\mathcal{H}terms(l\sigma, f) = \{d, e\} \\
\{a,b\} \cap_{|AC} \{d,e\} = \emptyset
\end{array} \right.
$$

We can see intuitively that the deduced clause $(f(c, f(d,e)) \simeq f(g, f(a,b)))$ *rewrites into a tautology, by applying two AC-paramodulation steps from* $(f(a,b) \simeq c)$ *and* $(f(d,e) \simeq g)$*. Moreover, each inference step using* $(f(c, f(d,e)) \simeq f(g, f(a,b)))$ *can be replaced by inference step(s) using* $(f(a,b) \simeq c)$ *and/or* $(f(d,e) \simeq g)$ *instead.*

2. *If* $x\sigma$ *and* $y\sigma$ *have common* $\mathcal{H}terms$ *:*

$$
\frac{(f(a, f(b,g)) \simeq c) \qquad (f(f(g,d), a) \simeq e)}{(f(c, f(g,d)) \simeq f(e, f(b,g)))} \qquad \textit{where} \quad \sigma = \{x \leftarrow f(g,d), y \leftarrow f(b,g)\}
$$

$$
\textit{Here,} \qquad \left\{ \begin{array}{l}
\mathcal{H}terms(x\sigma, f) = \{g, d\} \\
\mathcal{H}terms(y\sigma, f) = \{b, g\} \\
\{g,d\} \cap_{|AC} \{b,g\} = \{g\}
\end{array} \right.
$$

We can apply another AC-extended paramodulation step between the same initial clauses, but with the substitution $\tau = \{x \leftarrow d, y \leftarrow b\}$ *, to get the clause* $(f(c,d) \simeq f(e,b))$*.*

Then, using this clause $(f(c,d) \simeq f(e,b))$*, we can see that the first clause* $(f(c,f(g,d)) \simeq f(e,f(b,g)))$ *rewrites into a tautology by an AC-contextual paramodulation step. Moreover, each inference step using the first deduction* $(f(c,f(g,d)) \simeq f(e,f(b,g)))$ *can be replaced by inference step(s) with the second one* $(f(c,d) \simeq f(e,b))$*.*

Now that inference rules are defined, we have to prove the lifting lemma, since the proof of completeness will be done in the ground case.

5.2 Lifting Lemma

The purpose of the lifting lemma is to lift inferences from the ground level to the general level, i.e. to show that inferences made with ground clauses can be done with the corresponding general clauses. However, lifting paramodulation rules is often difficult. Let us illustrate this problem by an example : let $P(x,x,c)$ be a clause C and $(c \simeq a)$ an equation, with $c >_p a$; considering the instance $P(c,c,c)$ of $P(x,x,c)$, the paramodulation from $(c \simeq a)$ into the third argument of $P(c,c,c)$ produces $P(c,c,a)$; an analogous paramodulation into C generates $P(x,x,a)$, which admits $P(c,c,a)$ as an instance. However, if we paramodulate in the first argument of $P(c,c,c)$, we obtain $P(a,c,c)$. Since the paramodulation rule is never applied into a variable, there is no inference between C and $(c \simeq a)$ that can produce a clause which has $P(a,c,c)$ as an instance.

However, if we consider only substitutions which replace variables of $P(x,x,c)$ by irreducible terms, an instance of $P(x,x,c)$ in which we could paramodulate will be $P(a,a,c)$. Now, every paramodulation from $(c \simeq a)$ into $P(a,a,c)$ is an instance of a paramodulation from $(c \simeq a)$ into C. Hence the following definition :

Definition 17 *Let I be a (partial) E-interpretation and σ and θ two ground substitutions. We say that σ is I-reducible to θ, and we write* $\sigma \rightarrow_I \theta$ *, if σ is identical to θ except for a variable x, and* $\sigma(x) \longrightarrow_I \theta(x)$ *. If σ cannot be I-reduced to any substitution, we say that σ is I-irreducible.*

Theorem 18 (Irreducible Substitution Theorem) *Let I be a (partial) E-interpretation, and let $C\sigma$ be a clause whose atoms are all in the domain of I. Then, there exists a ground I-irreducible substitution θ such that* $I(C\sigma) = I(C\theta)$ *.*

The proof of this theorem is detailed in [HR91], and it is based on the noetherianity of the relation $\rightarrow_I$.

This theorem indicates that we only have to consider I-irreducible substitutions. It allows to lift paramodulation from ground clauses to clauses in general, by considering, in ground clauses, only those positions that already exist in the corresponding general clause.

First, we will use the following property, derived from the stability of orderings $\succ$ and $\succ_A$.

Proposition 19 *Let A and B be two objects (terms or atoms). If* $A\sigma \succ B\sigma$ *for a substitution σ, then* $A \not\preceq B$ *.*

A consequence of this proposition is :

Corollary 20 *Let σ be a ground AC-unifier of objects $A_1, \ldots, A_m$, and B_1 and B_2 two objects such that : $B_1\sigma \succ B_2\sigma$. There exists an AC-mgu τ of $A_1, \ldots, A_m$ such that : $B_1\tau \not\preceq B_2\tau$.*

This corollary permits to lift ordering conditions of inference rules. The substitution τ is defined by : $\sigma = \rho \circ \tau$, for a ground substitution ρ.

Lemma 21 (Lifting Lemma) *Let $C_1, \ldots, C_n$ be clauses and σ a ground substitution. If an inference rule R between $C_1\sigma, \ldots, C_n\sigma$ produces a clause C, the same inference rule R can be applied between clauses $C_1, \ldots, C_n$. It generates a clause D, which admits C as a ground instance.*

Proof : *The inference between $C_1\sigma, \ldots, C_n\sigma$ uses an AC-unifier of objects $A_1\sigma, \ldots, A_m\sigma$. But, since σ is ground, these objects are AC-equal. Let τ be an AC-mgu of $A_1, \ldots, A_m$ such that $\sigma = \rho \, o \, \tau$, for a ground substitution ρ. By Corollary 20, ordering conditions required by the rule remain valid, when applying τ instead of σ on terms and atoms.*

If R is a paramodulation or a contextual paramodulation, the position p exists in the general clause, since σ is assumed to be irreducible.

The last conditions which have to be satisfied by τ are the special conditions of contextual and extended paramodulation rules.

- *<u>Contextual Paramodulation :</u> Let $C_1 = (s \simeq t) \vee C_1'$ and $C_2 = L \vee C_2'$. The contextual paramodulation step is applied at the position p of L. We have to prove the property :*

  $$\forall w \in \mathcal{H}terms(L_{|_p}, f) \ , \ \ \mathcal{H}terms(s\tau, f) \ \not\subseteq_{AC} \ \mathcal{H}terms(w\tau, f)$$

  If such a term w existed, it would be a variable, and the term associated to w by τ (and also σ) contains $s\tau$ and is also reducible. So, it contradicts the condition of irreducibility of σ. Hence, there is no such term w.

  We can check that $s\tau$ is not a variable in the following way : if $s\tau = z \in \mathcal{X}$, as $f(z, x\tau) =_{AC} (L_{|_p})\tau$, it implies that there is a term w of $\mathcal{H}terms(L_{|_p}, f)$, such that $w\tau$ is either z or $f(z, w')$ for a term w', and also it contradicts previous property on $\mathcal{H}terms$.

- *<u>Extended Paramodulation :</u> Let $C_1 = (s \simeq t) \vee C_1'$ and $C_2 = (l \simeq r) \vee C_2'$. By definition of an extended paramodulation, we have :*

  $$\begin{cases} \mathcal{H}terms(s\sigma, f) \cap_{|AC} \mathcal{H}terms(l\sigma, f) \neq \emptyset & (1) \\ \mathcal{H}terms(x\sigma, f) \cap_{|AC} \mathcal{H}terms(y\sigma, f) = \emptyset & (2) \end{cases}$$

  Let us prove by refutation that $\mathcal{H}terms(x\tau, f) \cap_{|AC} \mathcal{H}terms(y\tau, f) = \emptyset$. We assume that there are two elements a and b, respectively of $\mathcal{H}terms(x\tau, f)$ and $\mathcal{H}terms(y\tau, f)$, such that $a =_{AC} b$. So, $a\rho =_{AC} b\rho$, and by definition of $\mathcal{H}terms$, $\mathcal{H}terms(a\rho, f) =_{AC} \mathcal{H}terms(b\rho, f)$. Hence, $\mathcal{H}terms(x\tau\rho, f) \cap_{|AC} \mathcal{H}terms(y\tau\rho, f) \neq \emptyset$. Since $\tau\rho = \sigma$, there is a contradiction with property (2). So, $\mathcal{H}terms(x\tau, f) \cap_{|AC} \mathcal{H}terms(y\tau, f) = \emptyset$.

  By definition of τ, we have : $f(s\tau, x\tau) =_{AC} f(l\tau, y\tau)$. So,

  $$\begin{aligned} &\mathcal{H}terms(s\tau, f) \cup_{|AC} \mathcal{H}terms(x\tau, f) \\ &\qquad =_{AC} \ \mathcal{H}terms(l\tau, f) \cup_{|AC} \mathcal{H}terms(y\tau, f) \end{aligned}$$

  We have proved that $\mathcal{H}terms(x\tau, f)$ and $\mathcal{H}terms(y\tau, f)$ admit no common elements. Hence,

  $$\begin{cases} \mathcal{H}terms(x\tau, f) \subseteq_{AC} \mathcal{H}terms(l\tau, f) \\ \mathcal{H}terms(y\tau, f) \subseteq_{AC} \mathcal{H}terms(s\tau, f) \end{cases}$$

  Assume that $\mathcal{H}terms(x\tau, f) \subset_{AC} \mathcal{H}terms(l\tau, f)$; all subterms of $l\tau$, which are not in $x\tau$, have also to be in $s\tau$. So, $\mathcal{H}terms(s\tau, f)$ and $\mathcal{H}terms(l\tau, f)$ possess common elements.

The second and last case is $\mathcal{H}terms(x\tau, f) =_{AC} \mathcal{H}terms(l\tau, f)$ *; it means that* $l\tau =_{AC} x\tau$ *and* $s\tau =_{AC} y\tau$ *, and therefore :* $l\sigma =_{AC} l\tau\rho =_{AC} x\tau\rho =_{AC} x\sigma$ *and* $s\sigma =_{AC} s\tau\rho =_{AC} y\tau\rho =_{AC} y\sigma$ *. The property (2) implies that* $\mathcal{H}terms(s\sigma, f) \cap_{|AC} \mathcal{H}terms(l\sigma, f) = \emptyset$ *, which is in contradiction with (1). So this case is impossible.*

We can easily check that neither $s\tau$ *nor* $l\tau$ *are variables ; indeed, assuming that* $s\tau$ *is a variable* z*, and since* $s\tau$ *and* $l\tau$ *admit common* $\mathcal{H}terms$*,* z *is in* $\mathcal{H}terms(l\tau, f)$ *; it implies that* $y\tau$ *is included in* $x\tau$ *; but, we have proved that* $x\tau$ *and* $y\tau$ *admit no common* $\mathcal{H}terms$*. This yields a contradiction. Therefore,* $\mathcal{H}ead(s\tau) = \mathcal{H}ead(l\tau) = f$*.*

So, all conditions have been verified, and we can apply the rule R *on clauses* $C_1, \ldots, C_n$*, to produce a clause* $D\tau$*. Since the ground clause* C *is equal to* $C'\sigma$*, and* $\sigma = \rho \circ \tau$*, the clause* $D\tau\rho$ *is equal to* $C'\sigma$*, and also to* C*. Hence,* C *is a ground instance of* $D\tau$*.* □

6 Refutational Completeness of AC Paramodulation

In this section, we shall prove the refutational completeness of the inference rules introduced in the previous section. The main differences between this proof and the proof in the empty theory are the construction of the rightmost maximal path in the semantic tree, and the numerous additional subcases introduced by the associative commutative axioms when considering failure nodes. We need to show that these additional failure nodes can be handled by our set of inference rules.

Let *INF* be the set of inference rules $\{$*AC-factoring, AC-reflexion, AC-resolution, AC-paramodulation, AC-contextual paramodulation, AC-extended paramodulation*$\}$. In the following proofs, we only consider inferences on ground instances of clauses of $INF^*(S)$. For the general case, lifting lemmas described in Section 5.2 can be applied, even for AC-paramodulation and AC-contextual paramodulation rules, since, by the Irreducible Substitution Theorem (theorem 18), it is always possible to label a failure node K by a clause whose variables are instantiated by K-irreducible terms ; hence, paramodulation can be restricted to non-variable positions. This argument is standard and will not be discussed further.

Theorem 22 (Completeness Theorem) *If* S *is an AC-unsatisfiable set of clauses, then* $INF^*(S)$ *contains the empty clause.*

The remaining of this section is devoted to the proof of this Completeness Theorem. We first give a sketch of it :

Sketch of Proof : *Let us assume that* $INF^*(S)$ *does not contain the empty clause, then,* $MCT(INF^*(S) \cup AC)$ *is non empty. We shall define (definition 25) by induction a sequence of nodes in* $MCT(INF^*(S) \cup AC)$*, called the quasi-rightmost path, and prove (proposition 27) that all these nodes are AC-consistent (definition 23), i.e. compatible with AC-axioms. In order to do this, we build the sequence so that it avoids what we call quasi-failure nodes (definition 24).*

The last node Q_γ *of the quasi-rightmost path is followed by failure and/or quasi-failure nodes in* $MCT(INF^*(S) \cup AC)$*. We shall show in proposition 35 that an inference step, using clauses labeling these nodes, can be applied, and that the deduced clause is falsified by* Q_γ*. This contradicts proposition 27.* □

6.1 Quasi-Failure Nodes and Quasi-Rightmost Path

To prove the Completeness Theorem, we reason by contradiction. We assume that S is an AC-unsatisfiable set of clauses and that $INF^*(S)$ does not contain the empty clause. Therefore $MCT(INF^*(S) \cup AC)$ is not empty. Let us define the sets

$$\begin{array}{rcl} \mathcal{GS} & = & \{ \ C \mid \exists C' \in INF^*(S) \ , \ C =_{AC} C'\sigma \ , \ \text{for some ground instance } C'\sigma \text{ of } C' \ \} \ , \\ \mathcal{AC} & = & \{ \ (u \simeq u') \mid u, u' \in \mathcal{T}(\mathcal{F}), \ u =_{AC} u', \ \mathcal{H}ead(u) \in \mathcal{F}_{AC} \ \} \ \text{ and} \\ \mathcal{S}^* & = & \mathcal{GS} \ \cup \ \mathcal{AC} \end{array}$$

$\mathcal{GS}$ is also the set of all clauses which are AC-equal to a ground instance of a clause of $INF^*(S)$, and $\mathcal{AC}$ is the set of all equalities which are AC-equal to a ground instance of an AC-axiom. By definition of the maximal consistent E-semantic tree, $MCT(\mathcal{S}^*)$ is equivalent to $MCT(INF^*(S) \cup AC)$. Hence, $MCT(\mathcal{S}^*)$ is non empty too.

In the following proofs, in general, an equality $(u \simeq v)$ will implicitly verify $u \geq v$.

We can always assume that there is a unique maximal literal in a ground clause, since the factoring rule allows us to eliminate multiple occurrences of this literal. We do not elaborate on this point, since it is quite similar to the standard case [HR91].

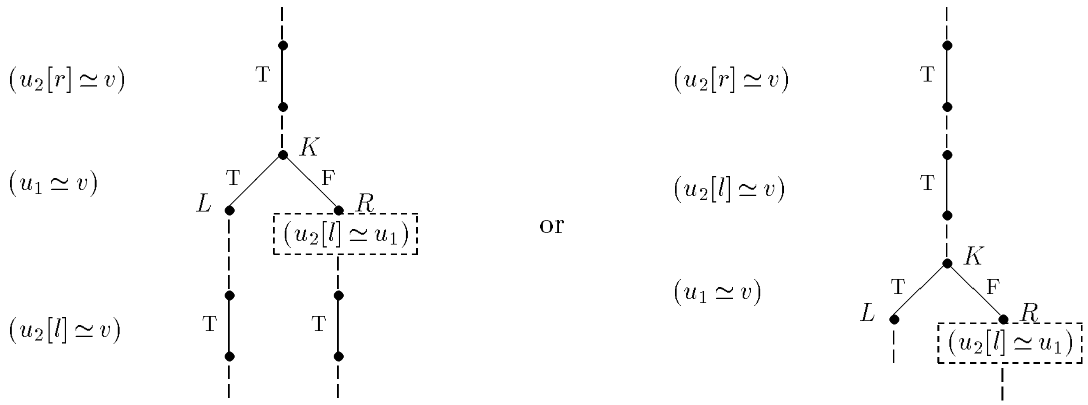
The method used for the empty theory is to build a sequence of partial E-interpretations, by transfinite induction, following the rightmost path of the maximal consistent E-semantic tree, then to prove that it is empty and derive a contradiction with the non-emptiness hypothesis of the tree. However, in the present case, the rightmost path may be an *AC-inconsistent path* and should not be considered. We shall instead build the rightmost AC-consistent path. First, let us define what we mean by an *AC-consistent node* in $MCT(\mathcal{S}^*)$.

Definition 23 *Let K be a partial E-interpretation of $MCT(\mathcal{S}^*)$, defined on W_α. K is AC-inconsistent, if :*

$$\exists B_1, B_2 \in W_\alpha \ , \ B_1 =_{AC} B_2 \quad \textit{and} \quad K(B_1) \neq K(B_2)$$

Otherwise, K is said to be AC-consistent.

We shall also need to define an extension of the notion of failure node, *quasi-failure node*, in order to detect an AC-inconsistency as soon as it occurs on a partial E-interpretation. Since an AC-inconsistency may be well-hidden, the notion of quasi-failure node is rather tricky. It can be illustrated by the following figures :



Definition 24 *Let K be a node of $MCT(\mathcal{S}^*)$, defined on W_α, such that A_α is an atom $(u_1 \simeq v)$, with $u_1 \succ v$ and K admits two extensions L and R, with $L(A_\alpha) = T$, $R(A_\alpha) = F$, and $R \in MCT(\mathcal{S}^*)$. Then, R is a quasi-failure node if there is a term $u_2[l]$, AC-equal to u_1, which is K-reducible by a K-irreducible atom $(l \simeq r)$ to $u_2[r]$, with $l \succ r$, and $K(u_2[r] \simeq v) = T$. The label of the quasi-failure node is defined to be $(u_1 \simeq u_2[l])$.*

In the following, when there will be no ambiguity, at a Q_α-irreducible atom A_α, the right extension of Q_α will be called *right node of A_α*. For instance, in previous figures, R is the right node of $(u_1 \simeq v)$.

We define now the *quasi-rightmost path* of $MCT(\mathcal{S}^*)$ as the rightmost path of $MCT(\mathcal{S}^*)$ which does not contain a quasi-failure node. Then, we shall prove that this path defines an AC-consistent partial E-interpretation.

Definition 25 *The quasi-rightmost path of $MCT(\mathcal{S}^*)$ is the partial E-interpretation Q_γ, defined on W_γ, and is built as follows : Q_0 is the empty interpretation ; we assume that Q_i has been defined for all $i < \alpha$; then, we extend the path to Q_α, if possible, by :*

- *If α is a limit ordinal, as in the classical case [HR91], we simply define Q_α by $\bigcup_{i<\alpha} Q_i$. Then, by the Closure Lemma, Q_α belongs to $MCT(\mathcal{S}^*)$.*
- *If α is not a limit ordinal, then α has a predecessor α^-. Several cases may occur :*
  - *(a) If Q_{α^-} has no successor in $MCT(\mathcal{S}^*)$, then we take $\gamma = \alpha^-$*
  - *(b) If Q_{α^-} has exactly one successor J in $MCT(\mathcal{S}^*)$, then $Q_\alpha = J$*
  - *(c) If Q_{α^-} has exactly two successors L and R, at least one of them is in $MCT(\mathcal{S}^*)$, $L(A_{\alpha^-}) = T$ and $R(A_{\alpha^-}) = F$, then*
    - *i. If R is a quasi-failure or a failure node, then*
      - *A. If L is a failure node, then $\gamma = \alpha^-$*
      - *B. If L is not a failure node, then $Q_\alpha = L$*
    - *ii. If R is neither a quasi-failure nor a failure node, then $Q_\alpha = R$*

The next proposition shows that, if an atom A_α is Q_α-reducible by an equality $(l \simeq r)$, with $l \succ r$, then there is an AC-equivalent atom A_β which is Q_α-reducible by an equality $(g \simeq d)$ whose right node is a failure node labeled by a clause of $\mathcal{GS}$. This proposition will simplify the proofs of the following propositions.

Proposition 26 *Let Q_α be the restriction of the quasi-rightmost path Q_γ to W_α. If there is a Q_α-irreducible atom $(l \simeq r)$ such that $l \succ r$ and $A_\alpha \longrightarrow_{Q_\alpha}^{l \simeq r} A_\alpha[r]$, then there are two atoms A_β and $(g \simeq d)$, such that $A_\beta =_{AC} A_\alpha$, $g \succ d$, $(g \simeq d)$ is Q_α-irreducible, $A_\beta \longrightarrow_{Q_\alpha}^{g \simeq d} A_\beta[d]$, $Q_\alpha(A_\beta[d]) = Q_\alpha(A_\alpha[r])$, and the right node of $(g \simeq d)$ is a failure node labeled by a clause of $\mathcal{GS}$.*

Proof : *Suppose that the proposition is true for all atoms less than but not AC-equal to A_α. Since $(l \simeq r)$ is Q_α-irreducible, it admits two extensions. Let R_l be the right node of $(l \simeq r)$: $R_l(l \simeq r) = F$.*

- If R_l is a failure node, then it cannot falsify a clause of $\mathcal{AC}$, since $l \succ r$ implies that $l \neq_{AC} r$. Thus, in this case, we can take $(l \simeq r)$ for $(g \simeq d)$ and A_α for A_β.

- If R_l is a quasi-failure node, it is labeled by an atom $(l \simeq l'[a])$; $(l'[a] \simeq r)$ is reducible by

an irreducible equality $(a \simeq b)$ *to* $(l'[b] \simeq r)$, *where* $a \succ b$, *and* $R_l(l'[b] \simeq r) \simeq Q_\alpha(l'[b] \simeq r) = T$. *So, the atom* $A_\alpha[l'[a]]$, *which is AC-equal to* $A_\alpha[l]$, *is* Q_α*-reducible by* $(a \simeq b)$ *to* $A_\alpha[l'[b]]$, *and* $Q_\alpha(A_\alpha[l'[b]]) = Q_\alpha(A_\alpha[r])$ *since* $A_\alpha[l'[a]]$ *is* Q_α*-reducible by* $(l'[a] \simeq r)$ *to* $A_\alpha[r]$.
By hypothesis, since $(l'[a] \simeq r) \prec_A A_\alpha$, *we can assume that there are two atoms* $(l''[a'] \simeq r)$ *and* $(a' \simeq b')$ *such that* $l''[a'] =_{AC} l'[a]$, $a' \succ b'$, $(a' \simeq b')$ *is* Q_α*-irreducible,* $(l''[a'] \simeq r) \longrightarrow_{Q_\alpha}^{a' \simeq b'} (l''[b'] \simeq r)$, $Q_\alpha(l''[b'] \simeq r) = Q_\alpha(l'[b] \simeq r) = T$, *and the right node of* $(a' \simeq b')$ *is a failure node labeled by a clause of* $\mathcal{GS}$.
So, we can take $(a' \simeq b')$ *for* $(g \simeq d)$, *and* $A_\alpha[l''[a']]$ *for* A_β. □

6.2 The Quasi-rightmost Path is AC-consistent

Proposition 27 *If* $MCT(\mathcal{S}^*)$ *is non empty, then the quasi-rightmost path* Q_γ *of* $MCT(\mathcal{S}^*)$ *is non empty and AC-consistent.*

Proof : *First,* Q_γ *is non empty : by construction of the semantic tree,* A_0 *is an atom* $(a \simeq a)$ *and* $Q_1(A_0) = T$ *; so,* Q_1 *cannot be a failure or a quasi-failure node.*

Second, we have to prove that Q_γ *is AC-consistent. We reason by contradiction : if* Q_γ *is AC-inconsistent, there exists a minimal ordinal* α *such that* Q_α *is AC-inconsistent. Necessarily,* $\boldsymbol{\alpha}$ **is not a limit ordinal.** *Otherwise, as in the classical case [HR91], if* α *is a limit ordinal,* Q_α *is defined by* $\bigcup_{i<\alpha} Q_i$, *and some* Q_i *is AC-inconsistent ; this is impossible since* α *is minimal.*

Since α *is not a limit ordinal,* α *has a predecessor* α^-. *Let* K *be the partial E-interpretation* Q_{α^-}, *and let* B *denote* A_{α^-}. *By minimality of* α, K *is AC-consistent. Then,*

$$\exists A_\beta <_A B \ , \ A_\beta =_{AC} B \ \text{ and } \ Q_\alpha(B) \neq Q_\alpha(A_\beta)$$

We prove in lemma 28 that B *is an equality* $(u_2 \simeq v)$ *and* A_β *is an equality* $(u_1 \simeq v)$, *where* $u_1 =_{AC} u_2$ *and* $u_2 \succ v$. *From now on, we assume that* $A_\beta = (u_1 \simeq v)$ *is the smallest atom such that* $A_\beta =_{AC} B$ *and* $Q_\alpha(A_\beta) \neq Q_\alpha(B)$. *Then, several cases are possible, and the next lemmas show that they all yield a contradiction :*

1. *If* K *has exactly one successor,* Q_α *: a contradiction is derived by lemmas 29, 31, 32.*
2. *If* K *has two successors* L *and* R, *at least one of them is in the quasi-rightmost path* (Q_α), $L(B) = T$ *and* $R(B) = F$.
   (a) *If* Q_α *is* L *: a contradiction is derived by lemma 33.*
   (b) *If* Q_α *is* R *: a contradiction is derived by lemma 34.*

Since all cases are impossible, Q_α *cannot belong to the quasi-rightmost path* Q_γ, *and also this sequence of partial E-interpretations is AC-consistent.* □

Lemma 28 B *is an equality* $(u_2 \simeq v)$ *and* A_β *is an equality* $(u_1 \simeq v)$, *where* $u_1 =_{AC} u_2$ *and* $u_2 \succ v$.

Proof : *We decompose the proof of this lemma in three facts :*

1. $\boldsymbol{\mathcal{H}ead(B)}$ **is "$\simeq$"** *; otherwise* $B = P(s_1, \ldots, s_n)$, $A_\beta = P(t_1, \ldots, t_n)$, $s_i =_{AC} t_i$ *for each* i, *and there is a* j *such that* $K(s_j \simeq t_j) = F$ *; so,* K *should be AC-inconsistent.*
   Thus, in the following, we can assume : $\boldsymbol{B} = (\boldsymbol{u}_2 \simeq \boldsymbol{v}_2)$ *and* $\boldsymbol{A}_\beta = (\boldsymbol{u}_1 \simeq \boldsymbol{v}_1)$, *with* $u_2 \geq v_2$, $u_1 \geq v_1$, $u_1 =_{AC} u_2$ *and* $v_1 =_{AC} v_2$.

2. $\boldsymbol{u}_2 \neq_{AC} \boldsymbol{v}_2$; *indeed, assuming that* $Q_\alpha(B) = F$, *if* $u_2 =_{AC} v_2$, *either* $(u_2 \simeq v_2) \in \mathcal{AC}$ *(if* $\mathcal{H}ead(u_2) \in \mathcal{F}_{AC}$ *), or there is a position in* u_2 *such that* $u_{2|_p} =_{AC} v_{2|_p}$, $\mathcal{H}ead(u_{2|_p}) \in \mathcal{F}_{AC}$ *(i.e.* $(u_{2|_p} \simeq v_{2|_p}) \in \mathcal{AC}$ *) and* $K(u_{2|_p} \simeq v_{2|_p}) = F$. *In both cases, an equation of* $\mathcal{AC}$ *is falsified by* K*, and also* K *should not belong to* $MCT(\mathcal{S}^*)$.
   If $Q_\alpha(B) = T$, *the reasoning is similar on* $(u_1 \simeq v_1)$*, since in that case* $Q_\alpha(A_\beta) = F$.
   So, $u_2 \succ v_2$, *and by AC-compatibility,* $u_1 \succ v_1$.

3. **We can choose $\boldsymbol{A}_\beta$ such that $\boldsymbol{v}_1 = \boldsymbol{v}_2$** , *since if* $v_1 \neq v_2$, $K(v_1 \simeq v_2) = T$, $(u_1 \simeq v_1) \longrightarrow_K^{v_1 \simeq v_2} (u_1 \simeq v_2)$ *and* $Q_\alpha(B) \neq K(u_1 \simeq v_2)$; *then, we can take* $(u_1 \simeq v_2)$ *for* A_β*. Indeed,* K *is AC-consistent means that* B *cannot be* K*-reduced to* $(u_2 \simeq v_1)$ *by* $(v_2 \simeq v_1)$ $(K(u_1 \simeq v_1) = K(u_2 \simeq v_1))$*, and also* $v_1 > v_2$.

So, B *is an equality* $(u_2 \simeq v)$ *and* A_β *is an equality* $(u_1 \simeq v)$ *such that :* $u_1 =_{AC} u_2$ *and* $u_2 \succ v$.
□

Lemma 29 *If* K *has exactly one successor* (Q_α)*, then* $B = (u_2 \simeq v) \longrightarrow_K^{l \simeq r} (u_2[r] \simeq v)$ *with* $l \succ r$ *and* $(l \simeq r)$ *is* K*-irreducible.*

Proof : *In the following, we prove that* $B = (u_2 \simeq v)$ *is* K*-reducible by an equality* $(l \simeq r)$*,* l *is not AC-equal to* r*, and* l *is a subterm of* u_2*.*

1. **$\boldsymbol{B}$ is $\boldsymbol{K}$-reducible by some $(\boldsymbol{l} \simeq \boldsymbol{r})$** ; *otherwise, by construction of the tree,* $B = (u \simeq u)$ *and* $Q_\alpha(B) = T$, *i.e.* $u_2 = u = v$. *But,* $u_2 \succ v$ *implies that* $u_2 \neq v$. *Then, there is a* K*-irreducible equality* $(l \simeq r)$*, with* $l > r$ *such that :* $B \longrightarrow_K^{l \simeq r} B[r]$.

2. $\boldsymbol{l} \neq_{AC} \boldsymbol{r}$; *else,* K *should be AC-inconsistent, since* $K(A_\beta) \neq K(B[r])$ *and these atoms are AC-equal. So :* $B \longrightarrow_K^{l \simeq r} B[r]$ *with* $l \succ r$.

3. **$\boldsymbol{B}$ is $\boldsymbol{K}$-reducible at a position in $\boldsymbol{u}_2$** ; *indeed, if* $B \longrightarrow_K^{l \simeq r} (u_2 \simeq v[r])$ *, then* $A_\beta \longrightarrow_K^{l \simeq r} (u_1 \simeq v[r])$; *since* $K(u_2 \simeq v[r]) \neq K(u_1 \simeq v[r])$, K *should be AC-inconsistent.*

Thus, we have proved that $B \longrightarrow_K^{l \simeq r} (u_2[r] \simeq v)$ *with* $l \succ r$. □

The following lemma will be used to prove lemmas 31, 32 and 33. It states that, for a node K of the quasi-rightmost path, if two AC-equal atoms, taking a different value for K, are K-reducible by different equalities at different positions, then a failure node should occur earlier in the path, provided that K is AC-consistent.

Lemma 30 *Let* K *be a node of the quasi-rightmost path* Q_γ*. Let* $(u_1 \simeq v)$ *and* $(u_2 \simeq v)$ *be atoms such that :*

- $u_1 =_{AC} u_2$, $u_1 \succ v$ *(and also* $u_2 \succ v$ *)*
- $(u_1 \simeq v) \longrightarrow_K^{g \simeq d} (u_1[d] \simeq v)$, $g \succ d$
- $(u_2 \simeq v) \longrightarrow_K^{l \simeq r} (u_2[r] \simeq v)$, $l \succ r$
- $(g \simeq d)$ *and* $(l \simeq r)$ *are* K*-irreducible*
- $K(u_1[d] \simeq v) \neq K(u_2[r] \simeq v)$

If K *is AC-consistent, then* K *falsifies a clause of* $\mathcal{GS}$*.*

Proof : *Let* $(g \simeq d)$*,* $(l \simeq r)$*,* $(u_1 \simeq v)$ *and* $(u_2 \simeq v)$ *be the atoms described just above, and let us assume that* K *is AC-consistent.*

By proposition 26, there are atoms $(l_1 \simeq r_1)$ *and* $(w_1[l_1] \simeq v)$ *such that :*

- $w_1[l_1] =_{AC} u_1$
- $(l_1 \simeq r_1)$ *is* K*-irreducible*
- $(w_1[l_1] \simeq v) \longrightarrow_K^{l_1 \simeq r_1} (w_1[r_1] \simeq v)$, $l_1 \succ r_1$
- *the right node of* $(l_1 \simeq r_1)$ *is a failure node labeled by a clause* C_1 *of* $\mathcal{GS}$
- $K(u_1[d] \simeq v) = K(w_1[r_1] \simeq v)$

In the same way, there are atoms $(l_2 \simeq r_2)$ *and* $(w_1[l_1] \simeq v)$ *such that :*

- $w_2[l_2] =_{AC} u_2$
- $(l_2 \simeq r_2)$ *is* K*-irreducible*
- $(w_2[l_2] \simeq v) \longrightarrow_K^{l_2 \simeq r_2} (w_2[r_2] \simeq v)$, $l_2 \succ r_2$
- *the right node of* $(l_2 \simeq r_2)$ *is a failure node labeled by a clause* C_2 *of* $\mathcal{GS}$
- $K(u_2[r] \simeq v) = K(w_2[r_2] \simeq v)$

Since $K(u_1[d] \simeq v) \neq K(u_2[r] \simeq v)$, $K(w_1[r_1] \simeq v) \neq K(w_2[r_2] \simeq v)$ *and* $K(w_1[r_1] \simeq w_2[r_2]) = F$ *. From this, we can deduce the following facts :*

1. $\boldsymbol{l_1}$ **and** $\boldsymbol{l_2}$ **are at AC-dependent positions** *in the AC-congruence class of* w_1*, i.e. there is no term* w*, AC-equal to* w_1*, such that :* $w = w[l_1]_{p_1}[l_2]_{p_2}$, $w[r_1]_{p_1}[l_2]_{p_2} =_{AC} w_1[r_1]$ *and* $w[l_1]_{p_1}[r_2]_{p_2} =_{AC} w_2[r_2]$ *. Indeed,* $K(w[l_1]_{p_1}[r_2]_{p_2} \simeq w[r_1]_{p_1}[l_2]_{p_2}) = T$ *since it is* K*-reducible to* $(w[r_1]_{p_1}[r_2]_{p_2} \simeq w[r_1]_{p_1}[r_2]_{p_2})$ *and AC-equal to* $(w_1[r_1] \simeq w_2[r_2])$*, and these atoms have a different value for* K*. It would mean that* K *is AC-inconsistent.*

2. $\boldsymbol{l_1}$ **and** $\boldsymbol{l_2}$ **are not AC-equal**. *Otherwise,* $K(l_1 \simeq r_2) = K(l_2 \simeq r_1) = T$, $(l_1 \simeq r_1)$ *is not reducible by* $(l_1 \simeq r_2)$ *implies that* $r_1 \leq r_2$ *, and* $(l_2 \simeq r_2)$ *is not reducible by* $(l_2 \simeq r_1)$ *implies that* $r_1 \geq r_2$ *. So,* $r_1 = r_2$ *. But,* $w_1[r_1] =_{AC} w_2[r_2]$ *and* $K(w_1[r_1] \simeq w_2[r_2]) = F$ *would mean that* K *is AC-inconsistent.*

3. $\boldsymbol{l_1}$ **is not a proper subterm of a** $\boldsymbol{l_3}$**, AC-equal to** $\boldsymbol{l_2}$ *; otherwise,* $(l_3 \simeq r_2)$ *would be* K*-reducible by* $(l_1 \simeq r_1)$*, and also the failure node at the right node of* $(l_2 \simeq r_2)$ *would be a quasi-failure node, labeled by the atom* $(l_2 \simeq l_3)$ *instead of a clause of* $\mathcal{GS}$*.*

4. *In the same way,* $\boldsymbol{l_2}$ **is not a proper subterm of a** $\boldsymbol{l_3}$**, AC-equal to** $\boldsymbol{l_1}$.

These four facts imply that there is a position q_1 *of* w_1 *and* q_2 *of* w_2 *such that :* ${w_1}_{|q_1} =_{AC} f(l_1, t_1)$ $=_{AC} {w_2}_{|q_2} =_{AC} f(l_2, t_2)$ *, where* f *is the AC-operator at the head of* l_1 *and* l_2*. Since* l_1 *and* l_2 *are at dependent positions, but none of them is a subterm of the other, there are ground terms* a*,* b *and* c *such that :* $l_1 =_{AC} f(a, b)$, $l_2 =_{AC} f(a, c)$ *, where :*

$$\begin{aligned} \mathcal{H}terms(l_1, f) \cap_{|AC} \mathcal{H}terms(l_2, f) &= \mathcal{H}terms(a, f) \\ \mathcal{H}terms(b, f) \cap_{|AC} \mathcal{H}terms(c, f) &= \emptyset \end{aligned}$$

We can notice that t_1 *and* t_2 *are respectively* c *and* b*, or* $f(c, d)$ *and* $f(b, d)$ *for a ground term* d*. Let* C_1 *and* C_2 *be the clauses of* $\mathcal{GS}$ *labeling the right nodes of* $(l_1 \simeq r_1)$ *and* $(l_2 \simeq r_2)$ *respectively. Then, denoting* $C'_1 = (l'_1 \simeq r'_1) \vee D'_1$ *and* $C'_2 = (l'_2 \simeq r'_2) \vee D'_2$ *the clauses of* $INF^*(S)$ *such that* $C_1 =_{AC} C'_1\sigma$ *and* $C_2 =_{AC} C'_2\sigma$ *for a ground substitution* σ*, all conditions are satisfied to apply an* **AC-extended paramodulation** *step between clauses* $C'_1\sigma$ *and* $C'_2\sigma$*. The deduced clause* C *is* $(f(r'_1\sigma, c) \simeq f(r'_2\sigma, b)) \vee D'_1\sigma \vee D'_2\sigma$ *, which also belongs to* $\mathcal{GS}$*.*

Let us denote $wrq1 = w_1[r_1]_{|q_1}$ *and* $wrq2 = w_2[r_2]_{|q_2}$ *. Since* $wrq1 =_{AC} f(r_1, t_1)$ *and* $wrq2 =_{AC} f(r_2, t_2)$ *, and by AC-consistency of* K*,* $K(wrq1 \simeq f(r_1, t_1)) = K(wrq2 \simeq f(r_2, t_2)) = T$ *. If we do the hypothesis that* $K(f(r'_1\sigma, c) \simeq f(r'_2\sigma, b)) = T$ *, we have :*

$$K(f(r_1, c) \simeq f(r_2, b)) = K(f(r_1, t_1) \simeq f(r_2, t_2)) = K(wrq1 \simeq wrq2) = T$$

Then, $(w_1[r_1] \simeq w_2[r_2])$ is K-reducible to $(w_1[r_1] \simeq w_2[wrq1]_{q_2})$ or $(w_1[wrq2]_{q_1} \simeq w_2[r_2])$, whether $wrq1$ is smaller or bigger than $wrq2$. But, both sides of the deduced atom are AC-equal ; it implies that K is AC-inconsistent.
The hypothesis that K satisfies $(f(r'_1\sigma, c) \simeq f(r'_2\sigma, b))$ was wrong, and we conclude that K falsifies the clause C of $\mathcal{GS}$, produced by the AC-extended paramodulation step described above, and this E-interpretation is not in $MCT(\mathcal{S}^*)$. □

Lemma 31 *If K has exactly one successor (Q_α), then A_β is K-irreducible.*

Proof : *In lemma 29, we have proved that $B = (u_2 \simeq v) \longrightarrow_K^{l \simeq r} (u_2[r] \simeq v)$ with $l \succ r$ and $(l \simeq r)$ K-irreducible. Then, we will focus on A_β, proving first that it is K-irreducible into an AC-equal atom, second that it is K-irreducible by any other equality.*

1. ***A_β* is not *K*-reducible into an AC-equal atom** *; indeed, if it was K-reducible by an equality $(g \simeq d)$, where $g =_{AC} d$, $A_\beta[d]$ would be AC-equal to B and smaller than A_β. It is impossible since A_β has been chosen minimal (in proposition 27).*
2. ***A_β* is *K*-irreducible**. *If not, there is an equality $(g \simeq d)$ which K-reduces $(u_1 \simeq v)$ in $(u_1[d] \simeq v)$, where $g \succ d$ and $(g \simeq d)$ is K-irreducible. Since K is assumed to be AC-consistent, lemma 30 allows us to say that K falsifies a clause of $\mathcal{GS}$. Also, K cannot be a node of the quasi-rightmost path.*

The only valid solution is that A_β has to be K-irreducible. □

Lemma 32 *If K has exactly one successor (Q_α), then $K(A_\beta) = F$ and Q_α cannot be a node of the quasi-rightmost path.*

Proof : *From lemmas 29 and 31, we know that :*

$$B = (u_2 \simeq v) \longrightarrow_K^{l \simeq r} (u_2[r] \simeq v) \ \text{ with } \ l \succ r$$
$$A_\beta = (u_1 \simeq v) \ \text{ and } \ (l \simeq r) \text{ are } K\text{-irreducible } \ (u_1 =_{AC} u_2)$$

We will prove that a quasi-failure node should occur at A_β, and also that Q_α cannot belong to the quasi-rightmost path. But first, let us prove that $K(A_\beta) = F$.
If we assume that $K(A_\beta) = T$, the right node R of A_β, right extension of Q_β and falsifying a clause C_1 of $\mathcal{S}^*$, is

1. *either a failure node : $C_1 = (u_1 \simeq v) \vee D_1 \ \in \mathcal{GS}$ (since $u_1 \succ v$) ; thus, $(u_2 \simeq v) \vee D_1$ is in $\mathcal{GS}$ too, and is falsified by Q_α, which also should not be in $MCT(\mathcal{S}^*)$.*
2. *or a quasi-failure node : $C_1 = (u_1 \simeq u_3)$, $u_1 =_{AC} u_3$, and $(u_3 \simeq v)$ is K-reducible by a K-irreducible equality $(g \simeq d)$ $(g \succ d)$ to $(u_3[d] \simeq v)$. $R(u_3[d] \simeq v) = T$, and $R(u_1 \simeq v) = R(u_2 \simeq v) = Q_\alpha(u_2 \simeq v) = F$. Since we have two AC-equal atoms $(u_2 \simeq v)$ and $(u_3 \simeq v)$ which are K-reducible and have a different value for the E-interpretation $(K(u_2[r] \simeq v) \neq K(u_3[d] \simeq v))$, by the lemma 30, we can say that K falsifies a clause of $\mathcal{GS}$, and also it cannot belong to the quasi-rightmost path.*

Finally, $A_\beta = (u_1 \simeq v)$ is K-irreducible and $K(A_\beta) = F$; $B = (u_2 \simeq v)$ is K-reducible by an equality $(l \simeq r)$ to $(u_2[r] \simeq v)$, where $l \succ r$ and $Q_\alpha(B) = T$. Therefore, there should be a quasi-failure node at the right node of A_β, labeled by $(u_1 \simeq u_2)$, and Q_α cannot belong to the quasi-rightmost path. □

Lemma 33 *If K has two successors L and R, $L(B) = T$, $R(B) = F$, and Q_α is L, then Q_α cannot belong to the quasi-rightmost path.*

Proof : *If K has two successors L and R, $L(B) = T$ and $R(B) = F$, and $Q_\alpha = L$, then R is a quasi-failure or a failure node, labeled by a clause C_R. We know that : $B = (u_2 \simeq v)$, $A_\beta = (u_1 \simeq v)$, $u_1 =_{AC} u_2$, $u_1 \succ v$, $Q_\alpha(B) \neq K(A_\beta)\ (= F)$. Moreover, A_β is the smallest atom satisfying these conditions. Then, we can deduce the following facts :*

1. ***R* is not a failure node** *; otherwise, $C_R = B \vee D_R \in \mathcal{GS}$, $A_\beta \vee D_R$ is in $\mathcal{GS}$ too and it is falsified by K, which cannot be in $MCT(\mathcal{S}^*)$.*

   Since $u_2 \succ v$, R is a quasi-failure node, and $C_R = (u_2 \simeq u_3[l])$ ($u_2 =_{AC} u_3[l]$), $(u_3[l] \simeq v) \longrightarrow_K^{l \simeq r} (u_3[r] \simeq v)$ ($l \succ r$ and $(l \simeq r)$ is K-irreducible), $K(u_3[r] \simeq v) = T$.

2. **A_β is K-reducible**. *Indeed, $R(u_3[r] \simeq v) = T$ and $R(u_1 \simeq v) = R(u_2 \simeq v) = F$. But, $(u_1 \simeq v)$ is smaller than $(u_2 \simeq v)$, and if it was K-irreducible, a quasi-failure node should occur at its level, and also K should not be in $MCT(\mathcal{S}^*)$.*

   So : $A_\beta \longrightarrow_K^{g \simeq d} (u_1[d] \simeq v)$, with $g > d$ and $(g \simeq d)$ K-irreducible.

3. **$g \neq_{AC} d$**, *since A_β has been chosen minimal. Therefore, $g \succ d$.*

We can summarize previous facts by :

$$\begin{array}{l} (u_1 \simeq v) \longrightarrow_K^{g \simeq d} (u_1[d] \simeq v) \ \text{ with } \ g \succ d \\ (u_3 \simeq v) \longrightarrow_K^{l \simeq r} (u_3[r] \simeq v) \ \text{ with } \ l \succ r \\ K(u_1[d] \simeq v) \ \neq K(u_3[r] \simeq v) \end{array}$$

Then, by lemma 30, we can say that K falsifies a clause of $\mathcal{GS}$, and also it cannot belong to the quasi-rightmost path (Q_α too). □

Lemma 34 *If K has two successors L and R, $L(B) = T$, $R(B) = F$, and Q_α is R, then Q_α cannot belong to the quasi-rightmost path.*

Proof : *K has two successors L and R, $L(B) = T$ and $R(B) = F$, and $Q_\alpha = R$. We know that : $B = (u_2 \simeq v)$, $A_\beta = (u_1 \simeq v)$, $u_1 =_{AC} u_2$, $u_1 \succ v$, $Q_\alpha(B) \neq K(A_\beta)\ (= T)$. Moreover, A_β is the smallest atom satisfying these conditions. Then, we have to study two cases :*

- **If $(u_1 \simeq v)$ is K-reducible** *by a K-irreducible equality $(g \simeq d)$ to $(u_1[d] \simeq v)$: we can check that $(g \simeq d) \notin \mathcal{AC}$, since A_β has been chosen minimal ; so, $g \succ d$, and also R should be a quasi-failure node ($R(B) \neq R(A_\beta)$).*
- **If $(u_1 \simeq v)$ is K-irreducible** *: let C_1 be the clause falsified by the right node of $(u_1 \simeq v)$. First, we can check that $C_1 \notin \mathcal{GS}$, otherwise $C_1 = (u_1 \simeq v) \vee D_1$ and $(u_2 \simeq v) \vee D_1$ would be in $\mathcal{GS}$ too ; however, R falsifies this last clause, and also it could not be in $MCT(\mathcal{S}^*)$. Since $u_1 \succ v$, the right node of $(u_1 \simeq v)$ is a quasi-failure node, and C_1 is an atom $(u_1 \simeq u_3)$, where u_3 is AC-equal to u_1 and $(u_3 \simeq v) \longrightarrow_K^{l \simeq r} (u_3[r] \simeq v)$, where $l \succ r$ and $(l \simeq r)$ is K-irreducible. Moreover, $R(u_3[r] \simeq v) = T \neq R(u_2 \simeq v)$ implies that R should be a quasi-failure node too.*

Since the situation is impossible in both cases, Q_α cannot be a node of the quasi-rightmost path Q_γ. □

6.3 Proof of the Completeness Theorem

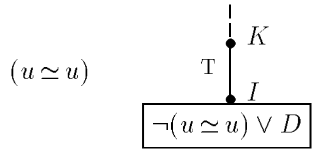
Since we have proved that Q_γ, the quasi-rightmost path of $MCT(\mathcal{S}^*)$ is non empty and AC-consistent, we will prove that Q_γ falsifies clauses of $\mathcal{GS}$. This will finish the proof of theorem 22, since, each node of the quasi-rightmost path being a node of $MCT(\mathcal{S}^*)$, if one of them falsifies a clause of $\mathcal{GS}$, it means that $MCT(\mathcal{S}^*)$ is empty, and also that the empty clause belongs to $\mathcal{GS}$, and hence to $INF^*(S)$.

Proposition 35 *The last node Q_γ of the quasi-rightmost path falsifies a clause of $\mathcal{GS}$.*

Proof : *Let $K = Q_\gamma$ be the last node of the quasi-rightmost path. Hence, K is defined on all the atoms A_i where $i < \gamma$, K belongs to* MCT($\mathcal{S}^*$)*, and every extension of K is a failure or a quasi-failure node. Let us write B for the atom A_γ.*
As in the empty theory [HR91], there are three main cases :

- either K has one extension I, and B is K-irreducible
- or K has two extensions L and R
- or K has one extension I, and B is K-reducible

<u>**Case 1 :**</u> *K has one extension I, and B is K-irreducible*



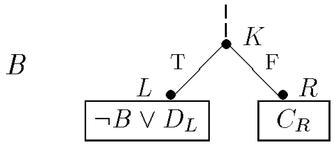
By definition of the construction of the semantic tree, the atom B is an equality of the form $(u \simeq u)$, which is K-irreducible, u being a ground term ($I(u \simeq u) = T$, by definition). Since I is a failure node for $\mathcal{S}^*$, there is a clause $C \in \mathcal{S}^*$ such that $I(C) = F$.
As K does not falsify C ($K \in MCT(\mathcal{S}^*)$), and K and I differ only by their value on B, we have :

$$C = \neg(u \simeq u) \vee D \quad \text{and} \quad K(D) = F \ ,$$

for a D whose atoms are smaller than $(u \simeq u)$.
As $I(u \simeq u) = T$, C belongs to $\mathcal{GS}$. So, there are a clause $C' \in INF^*(S)$ and a ground substitution σ such that $C'\sigma =_{AC} C$.
We can deduce the clause D by AC-reflexion of $C'\sigma$. Thus, D is in $\mathcal{GS}$.
Hence there is a contradiction between the fact that K belongs to $MCT(\mathcal{S}^*)$ and the fact that K falsifies D.

Therefore, first case cannot hold.

<u>**Case 2 :**</u> *K has two extensions L and R (thus B is K-irreducible)*



The node K has two extensions that are failure or quasi-failure nodes. And, since $L(B) = T$, there is a clause $C_L = \neg B \vee D_L$, where each atom of D_L is smaller than B (for the

ordering $\succ_A$*), such that* C_L *belongs to* $\mathcal{GS}$ *and* $L(C_L) = F$ *; just as* K *does not falsify* C_L *and* L *differs from* K *only by its value on* B*, we have :* $K(D_L) = L(D_L) = F$ *.*
Let us denote C'_L *the clause* $\neg B' \vee D'_L$ *of INF*$^*(S)$ *such that :* $C'_L\sigma =_{AC} C_L$ *, where* σ *is a ground substitution.*

As R *can be either a failure node or a quasi-failure node, we have to study two subcases.*

Case 2.1 : *If* R *is a failure node*

Then, C_R *is either in* $\mathcal{GS}$ *or in* $\mathcal{AC}$*.*

Case 2.1.1 : *If* $C_R \in \mathcal{GS}$

It means that there is a clause C'_R *of INF*$^*(S)$*, such that :* $C'_R\sigma =_{AC} C_R = B \vee D_R$ *. As for* D_L*, we have :* $K(D_R) = R(D_R) = F$ *.*
Hence, by AC-resolution between $C'_L\sigma$ *and* $C'_R\sigma$*, we can generate a clause AC-equal to* $D_L \vee D_R$ *; so, this last clause belongs to* $\mathcal{GS}$*.*
Hence K*, which falsifies* $D_L \vee D_R$*, cannot be in* $MCT(\mathcal{S}^*)$*.*

Case 2.1.2 : *If* $C_R \in \mathcal{AC}$

It means that $B \in \mathcal{AC}$*. So, we can apply an AC-reflexion step in* $C'_L\sigma$ *to deduce the clause* $D'_L\sigma$ *which is AC-equal to* D_L*. So,* D_L *belongs to* $\mathcal{GS}$ *and as it is falsified by* K*, this last E-interpretation cannot belong to* $MCT(\mathcal{S}^*)$*.*

Case 2.2 : *If* R *is a quasi-failure node*

In this case, by construction of the rightmost path, B *is of the form* $(u \simeq v)$*, with* $u \succ v$*, and* C_R *is* $(u \simeq u_1)$ *with* $u =_{AC} u_1$*.*
Then, the atom $(u_1 \simeq v)$ *is* K*-reducible by an irreducible atom* $(l \simeq r)$ $(l \succ r)$ *to* $(u_1[r] \simeq v)$*. Since* $l \neq_{AC} r$*, and by AC-compatibility of the ordering,* $(u_1[r] \simeq v)$ *is in the domain of* K *and :* $K(u_1[r] \simeq v) = T$*.*
Let β *be the index of* $(l \simeq r)$ *;* Q_β *is the restriction of* K *to* W_β*. As* $(l \simeq r)$ *is* Q_β*-irreducible,* Q_β *admits two extensions, and by proposition 26, we have proved that* M*, the right extension of* Q_β*, is a failure node labeled by a clause* C_M *of* $\mathcal{GS}$ *; let* C_M *be* $(l \simeq r) \vee D_M$ *; there is a clause* $C'_M = (l' \simeq r') \vee D'_M$ *of INF*$^*(S)$ *such that* C_M *is AC-equal to a ground instance* $C'_M\sigma$ *of* C'_M*. As* Q_β *and* M *differ only by their value on* $(l \simeq r)$*,* $Q_\beta(D_M) = M(D_M) = F$ $(= K(D_M))$ *.*
So, $B' = (u' \simeq v')$ *, where* $u'\sigma =_{AC} u$ *and* $v'\sigma =_{AC} v$ *, and there is an occurrence* $o \in \mathcal{MP}os(u'\sigma)$ *such that either* $u'\sigma_{|_o} =_{AC} l'\sigma$ *, or* $\mathcal{H}ead(l) = f \in \mathcal{F}_{AC}$ *and* $u'\sigma_{|_o} =_{AC} f(t, l'\sigma)$ *, where* t *is a ground term.*

- If $u'\sigma_{|_o} =_{AC} l'\sigma$ *, we can apply an AC-paramodulation step from* $C'_M\sigma$ *into* $C'_L\sigma$ *to get* $\neg(u'\sigma[o \leftarrow r'\sigma] \simeq v'\sigma) \vee D'_L\sigma \vee D'_M\sigma$ *.*

- If $u'\sigma_{|_o} =_{AC} f(t, l'\sigma)$ *, we can apply an AC-contextual paramodulation step from* $C'_M\sigma$ *into* $C'_L\sigma$ *to get* $\neg(u'\sigma[o \leftarrow f(t, r'\sigma)] \simeq v'\sigma) \vee D'_L\sigma \vee D'_M\sigma$ *. The condition on* $\mathcal{H}terms$ *for applying that inference step is trivial since we are using ground literals.*

In both cases, the deduced clause is AC-equal to $\neg(u_1[r] \simeq v) \vee D_L \vee D_M$ *. So, this last clause belongs to* $\mathcal{GS}$ *and is falsified by* K*. Hence, this last E-interpretation cannot belong to* $MCT(\mathcal{S}^*)$*.*

Case 2 is also impossible.

Case 3 : K *has one extension* I*, and* B *is* K*-reducible.*

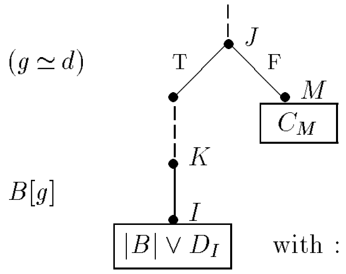
Let $(g \simeq d)$ *be the smallest equation such that* $g > d$ *,* $K(g \simeq d) = T$ *and there is an occurrence* p *such that*

$B_{|_p} = g$ *if* B *is not an equational literal or* $I(B) = F$,

$s_{|_p} = g$ *if* B *is an equation* $(s \simeq t)$, *with* $s > t$ *and* $I(B) = T$.

In the second case, the existence of $(g \simeq d)$ *is not immediate. It can be proved as follows : assume* s *is* K*-irreducible ; then* t *must be* K*-reducible to some* t' *; but then it is easy to show that* $K(s \simeq t') = T$ *and therefore* s *can be* K*-reduced by* $(s \simeq t')$, *yielding a contradiction.*

Let β *be the index of the atom* $(g \simeq d)$ *and* J *the restriction of* K *to* W_β *(defined on the atoms smaller than* $(g \simeq d)$*). We verify that* $(g \simeq d)$ *is* J*-irreducible. Hence* J *has two successors. Let* M *be the right successor of* J. M *is a failure or a quasi-failure node.*



with : $|B| =$ if $I(B) = F$ then B , else $\neg B$

By construction of the quasi-rightmost path, we can establish that I *falsifies a clause* $C_I = |B| \vee D_I$ *of* $\mathcal{GS}$, *where every atom of* D_I *is smaller than* B, *and* $I(D_I) = K(D_I) = F$ *; indeed,* I *cannot be a quasi-failure node, and if* C_I *was in* $\mathcal{AC}$, B *would be an equation* $(s[g] \simeq s')$, *where* $s[g] =_{AC} s'$, *and* K *would be* AC*-inconsistent, since* $K(s' \simeq s[d]) = K(s[g] \simeq s') = F$ *and* $K(s[g] \simeq s[d]) = K(s[d] \simeq s[d]) = T$.
Moreover, by definition of an E-interpretation, as $I(g \simeq d) = T$, *it implies that* B *and* $B[p \leftarrow d]$ *have the same truth value for* I. *Since* $B[d] <_A B$, *we can deduce that* $K(B[d]) = I(B[d])$.
Let $C'_I = |B'| \vee D'_I$ *denote the clause of* $INF^*(S)$ *verifying* $C'_I\sigma =_{AC} C_I$, *where* σ *is a ground substitution.*

As in Case 2, we have to consider two main subcases :

<u>Case 3.1 :</u> *If* M *is a failure node*

Then, C_M *is either in* $\mathcal{GS}$ *or in* $\mathcal{AC}$.

<u>Case 3.1.1 :</u> *If* $C_M \in \mathcal{GS}$

Then, $C_M = (g \simeq d) \vee D_M$, *every atom of* D_M *is less than* $(g \simeq d)$ *for the ordering* $\succ_A$, $J(D_M) = K(D_M) = F$, *and there is a clause* $C'_M = (g' \simeq d') \vee D'_M$ *of* $INF^*(S)$ *verifying* $C'_M\sigma =_{AC} C_M$. *Let* o *be an occurrence of* $B'\sigma$ *such that either* $B'\sigma_{|_o} =_{AC} g'\sigma$, *or* $\mathcal{H}ead(g) = f \in \mathcal{F}_{AC}$ *and* $B'\sigma_{|_o} =_{AC} f(t, g'\sigma)$ *for some ground term* t.

- If $B'\sigma_{|_o} =_{AC} g'\sigma$, *we can apply an AC-paramodulation step from* $C'_M\sigma$ *into* $C'_I\sigma$ *to get* $|B'\sigma[o \leftarrow d'\sigma]| \vee D'_I\sigma \vee D'_M\sigma$.

- If $B'\sigma_{|_o} =_{AC} f(t, g'\sigma)$, *we can apply an AC-contextual paramodulation step from* $C'_M\sigma$ *into* $C'_I\sigma$ *to get* $|B'\sigma[o \leftarrow f(t, d'\sigma)]| \vee D'_I\sigma \vee D'_M\sigma$.

In both cases, the generated clause is AC-equal to $|B[d]| \vee D_I \vee D_M$. *So this last clause belongs to* $\mathcal{GS}$.
As $K(|B[d]| \vee D_I \vee D_M) = F$, K *cannot be in* $MCT(\mathcal{S}^*)$.

Case 3.1.2 : *If* $C_M \in \mathcal{AC}$

It means that $g =_{AC} d$, *and also* $B[d] =_{AC} B$. *Hence,* $|B[d]| \vee D_I$ *belongs to* $\mathcal{GS}$ *too, and is falsified by* K. *So, a failure node occurs at the level of* $B[d]$, *and* K *is not in* $MCT(\mathcal{S}^*)$.

Case 3.2 : *If* M *is a quasi-failure node*

In this case, by definition of a quasi-failure node, $g \succ d$, *and* C_M *is* $(g \simeq g')$ *with* $g =_{AC} g'$. *Moreover,* $(g' \simeq d)$ *is reducible by an irreducible atom* $(l \simeq r)$ $(l \succ r)$ *to* $(g'[r] \simeq d)$, *and* $K(g'[r] \simeq d) = J(g'[r] \simeq d) = T$.

Let β *be the index of* $(l \simeq r)$ *and* G *be the restriction of* J *to* W_β. *As* $(l \simeq r)$ *is* G*-irreducible,* G *admits two extensions, and by proposition 26, we have proved that* H, *the right extension of* G, *is a failure node labeled by a clause* C_H *of* $\mathcal{GS}$ *; let* $(l \simeq r) \vee D_H$ *denote this clause* C_H *; there is a clause* $C'_H = (l' \simeq r') \vee D'_H$ *of INF*$^*(S)$ *such that* C_H *is AC-equal to a ground instance* $C'_H\sigma$ *of* C'_H. *As* G *and* H *differ only by their value on* $(l \simeq r)$, $G(D_H) = H(D_H) = F$ $(= K(D_H))$.

So, $B'\sigma =_{AC} B[g] =_{AC} B[g'[l]]$. *There is an occurrence* $o \in \mathcal{MP}os(B'\sigma)$ *such that either* $B'\sigma_{|_o} =_{AC} l'\sigma$, *or* $\mathcal{H}ead(l) = f \in \mathcal{F}_{AC}$ *and* $B'\sigma_{|_o} =_{AC} f(t, l'\sigma)$, *where* t *is a ground term.*

- If $B'\sigma_{|_o} =_{AC} l'\sigma$, *we can apply an AC-paramodulation step from* $C'_H\sigma$ *into* $C'_I\sigma$ *to get* $|B'\sigma[o \leftarrow r'\sigma]| \vee D'_I\sigma \vee D'_H\sigma$.

- If $B'\sigma_{|_o} =_{AC} f(t, l'\sigma)$, *we can apply an AC-contextual paramodulation step from* $C'_H\sigma$ *into* $C'_I\sigma$ *to get* $|B'\sigma[o \leftarrow f(t, r'\sigma)]| \vee D'_I\sigma \vee D'_H\sigma$.

In both cases, the deduced clause is AC-equal to $|B[g'[r]]| \vee D_I \vee D_H$. *By definition of an E-interpretation,* $K(g'[l] \simeq d) = K(g'[r] \simeq d) = T$ *; therefore,* $I(B[g'[r]]) = I(B[d])$, *and* $I(B[g'[r]]) = I(B[g])$. *Hence, the clause* $|B[g'[r]]| \vee D_I \vee D_H$ *is falsified by* K *and this interpretation cannot belong to* $MCT(\mathcal{S}^*)$.

Therefore Case 3 leads to a contradiction too.

Since all cases are impossible, necessarily $MCT(\mathcal{S}^*)$ *is empty and therefore the empty clause belongs to INF*$^*(S)$. $\square$

7 Reduction Rules

In this section, we introduce new inference rules that allow to delete redundant clauses, and therefore to reduce the search space of the theorem-proving procedures.

7.1 Subsumption and Simplification

Definition 36 (AC-subsumption) *Let* C_1 *and* C_2 *be two clauses satisfying :* C_2 *has at least as many literals as* C_1, *and there is a substitution* σ *such that* $C_1\sigma \subseteq_{AC} C_2$, *considering a clause as a multiset of literals. Then, we say that the clause* C_1 *AC-subsumes the clause* C_2.

Definition 37 (Strict AC-subsumption) *Let* C_1 *and* C_2 *be two clauses. We say that* C_1 *strictly AC-subsumes the clause* C_2 *if* C_1 *AC-subsumes* C_2 *and* C_2 *does not AC-subsumes* C_1.
The application of this rule consists of the deletion of the strictly subsumed clause (C_2).

The result of Loveland [Lov78] about the impossibility of an infinite sequence $C_0, C_1, \ldots$, where each C_{i+1} strictly subsumes C_i, for all i, remains true in AC theories :

Lemma 38 *There is no infinite sequence $C_0, C_1, \ldots$ such that the clause C_{i+1} strictly AC-subsumes the clause C_i, for all i.*

Now, we define two simplification rules, where AC-matching is used.

Definition 39 (AC-simplification)

$$\frac{(l \simeq r) \vee D_1 \qquad L \vee D_2}{L[r\sigma]_p \vee D_2} \quad \text{if} \quad \begin{cases} L_{|p} =_{AC} l\sigma \quad \textit{where} \quad p \in \mathcal{MP}os(L) \\ D_2 \;=\; D_2' \vee D_2'' \quad \textit{and} \quad D_1\sigma \;\subseteq_{AC}\; D_2'' \\ l\sigma \succ r\sigma \\ \exists A \in Atoms(L \vee D_2'),\; (l \simeq r)\sigma \prec_A A \end{cases}$$

Comments : *this ordered AC-simplification rule applies if there is a non variable occurrence p of the literal L and a match σ, such that $L_{|p}$ is AC-equal to $l\sigma$. Each literal of $D_1\sigma$ has to be AC-equal to a literal of D_2'' (and also of the same sign). Moreover, an atom of $L \vee D_2'$ has to be greater than $(l \simeq r)\sigma$ (it often is L itself).*
The application of this rule consists of the replacement of the reducible clause $L \vee D_2$ by the reduced one $L[r\sigma]_p \vee D_2$. The position p has to be a maximal occurrence in L of the head operator of $l\sigma$, if it is AC.

Definition 40 (AC-contextual simplification)

$$\frac{(l \simeq r) \vee D_1 \qquad L \vee D_2}{L[f(r,x)\sigma]_p \vee D_2} \quad \text{if} \quad \begin{cases} L_{|p} =_{AC} f(l,x)\sigma \\ \qquad \textit{where } f = \mathcal{H}ead(l\sigma) \in \mathcal{F}_{AC}, \\ \qquad\qquad x \textit{ is a new variable, } p \in \mathcal{MP}os(L) \\ D_2 \;=\; D_2' \vee D_2'' \quad \textit{and} \quad D_1\sigma \;\subseteq_{AC}\; D_2'' \\ l\sigma \succ r\sigma \end{cases}$$

Comments : *this ordered AC-contextual simplification rule applies if there is a position p of the literal L and a match σ, such that $L_{|p}$ is AC-equal to $f(l,x)\sigma$, where f is the head AC-operator of $l\sigma$ and p is a maximal occurrence of f in L. Each literal of $D_1\sigma$ has to be AC-equal to a literal of D_2'' (and also of the same sign). We can notice that : $L \succ_A (l \simeq r)\sigma$.*
The application of this rule consists of the replacement of the reducible clause $L \vee D_2$ by the reduced one $L[f(r,x)\sigma]_p \vee D_2$.

7.2 Completeness

Let INF_S be the set of inference rules : *AC-factoring, AC-reflexion, AC-resolution, AC-paramodulation, AC-contextual paramodulation, AC-extended paramodulation, AC-simplification, AC-contextual simplification* and *strict AC-subsumption.* A *derivation* from S_0 is a sequence $S_0, S_1, \ldots$ of sets of clauses obtained by application of inference rules of INF_S.
The method used in Section 6 cannot be used here since, if two clauses allow to deduce a third one, it could be possible that they never appear in a same set S_i ; so the inference will never be considered. We will show that, fortunately, this never keeps from generating the empty clause.

Definition 41 *A derivation $S_0, S_1, \ldots$ is fair if : if there is j such that $C \in \bigcap_{i \geq j} RP(S_i)$, then C is subsumed by $C' \in \bigcup_{i \geq 0} S_i$, where $RP(S_i)$ is the set of all clauses that can be inferred from S_i in one step of a rule of INF_S.*

From now on, we assume that S_0 is an AC-unsatisfiable set of clauses, and that $S_0, S_1, \ldots$ is a fair derivation. S^* denotes $\bigcup_{i \geq 0} S_i$.

Definition 42 *A clause C is persistent in a derivation $S_0, S_1, \ldots$ if there is an index k such that C belongs to $\bigcap_{i \geq k} S_i$.*

Let us define some notations :

$GR(S)$	:	set of clauses AC-equal to a ground instance of a clause of the set S, i.e. $\{ C \mid \exists C' \in S \ , \ C =_{AC} C'\sigma$, for some ground substitution $\sigma \}$

and for a failure node K :

Σ_K	:	set of clauses of S^* falsified by K, i.e. $\{ C \in S^* \mid \exists C' \in GR(\{C\}) \ , \ K(C') = F \}$
TG_K	:	set of minimal clauses of $GR(\Sigma_K)$ for the ordering $\succ_A$
Π_K	:	subset of clauses of Σ_K having an instance in TG_K (modulo AC)
Π'_K	:	set of minimal clauses of Π_K for the ordering of strict subsumption

We can notice that : $GR(\Pi_K) = GR(\Pi'_K) = TG_K$. Like to prove the Completeness Theorem (Theorem 22), we define the sets :

$$\begin{array}{rcl} \mathcal{GS} & = & GR(S^*) \\ \mathcal{AC} & = & \{ \ (u \simeq u') \mid u, u' \in \mathcal{T}(\mathcal{F}), \ u =_{AC} u', \ \mathcal{H}ead(u) \in \mathcal{F}_{AC} \ \} \\ \mathcal{S}^* & = & \mathcal{GS} \ \cup \ \mathcal{AC} \end{array}$$

The quasi-rightmost path Q_γ is built as described in Section 6.1.

Proposition 43 *Let K be a failure node, extension of an interpretation Q_α, restriction of Q_γ. We assume that K is either a right extension of Q_α or an extension of Q_γ. If Q_α is AC-consistent, K falsifies a persistent clause.*

Proof : *To prove that K falsifies a persistent clause, we will show that any element of Π'_K is persistent. So, let C' be a clause of Π'_K and C be the clause of TG_K, AC-equal to $C'\sigma$, ground instance of C' ; $K(C) = F$.*

1. *If C' is simplified :* *There is an index j such that : $C' \in S_j$ and a clause $(l' \simeq r') \vee D'$ of S_j simplifies C' at a non-variable position p of a literal L' of C', using the substitution ρ.*

   By definition of simplification rules, $C' = L' \vee C'_1 \vee C'_2$, where $D'\rho \subseteq_{AC} C'_2$. So, $C = L \vee C_1 \vee C_2$ where $L =_{AC} L'\sigma$, $C_1 =_{AC} C'_1\sigma$ and $D'\rho\sigma \subseteq_{AC} C_2 =_{AC} C'_2\sigma$.

   Since $L'_{|_p} =_{AC} l'\rho$ or $f(l', x)\rho$, there is a ground literal $L''[l'\rho\sigma]$ which is AC-equal to $L'\sigma$ (and also to L).

   By definition of simplification rules, there is an atom A in $L' \vee C'_1$ which is greater than $(l' \simeq r')\rho$ for the ordering $\succ_A$. By stability of this ordering, $A\sigma \succ_A (l' \simeq r')\rho\sigma$. So, $(l' \simeq r')\rho\sigma \vee C_2 \prec_A C$ and, since $K(C_2) = F$, $K((l' \simeq r')\rho\sigma)$ has to be true, otherwise C could not be in TG_K.

   Then, if $Q_\alpha(L''[r'\rho\sigma]) = F$, the clause $L''[r'\rho\sigma] \vee C_1 \vee C_2$ is falsified by K and is smaller than C for $\succ_A$. Moreover, it is AC-equal to the ground instance $L'[r'\rho]_p\sigma \vee C'_1\sigma \vee C'_2\sigma$ (or $L'[f(r', x)\rho]_p\sigma \vee C'_1\sigma \vee C'_2\sigma$) of the clause deduced by the simplification step (appearing in S_{j+1}). So, C cannot belong to TG_K.

If $Q_\alpha(L''[r'\rho\sigma]) = T$, L is this maximal literal of the clause C (A_α or $\neg A_\alpha$) since Q_α is AC-consistent.

(a) *If Q_α has two successors :*

i. *If K is the right extension of Q_α : K sould be a quasi-failure node, and not a failure node.*

ii. *If K is the left extension of Q_α : Then, the right extension R of Q_α is either a failure node or a quasi-failure node.*

A. *If R is a failure node : R labels a persistent clause C_R. $l'\rho \succ r'\rho$ implies by stability : $l'\rho\sigma \succ r'\rho\sigma$. By Proposition 26, there is an irreducible atom $(g \simeq d)$ (with $g \succ d$), whose right extension is labeled by a persistent clause C_g of $\mathcal{GS}$, which reduces an atom AC-equal to A_α. So, the clause E resulting from a paramodulation (or a contextual paramodulation) step from C_g into C_R belongs to $\bigcap_{i\geq j} RP(S_i)$ and is falsified by R. It should label a failure node cutting the quasi-rightmost path. That case is also impossible.*

B. *If R is a quasi-failure node : There is an atom A_β, AC-equal to A_α, which is reducible by an irreducible atom $(g \simeq d)$ with $g \succ d$. $Q_\alpha(A_\beta[d]) = T$. Moreover, we know that the atom $A_{\beta'}$, corresponding to the literal $L''[l'\rho\sigma]$, is reducible by the atom $(l'\rho\sigma \simeq r'\rho\sigma)$, and $l'\rho\sigma \succ r'\rho\sigma$. $Q_\alpha(A_{\beta'}[r'\rho\sigma]) = F$. By Lemma 30, a failure node should cut the quasi-rightmost path, which implies that this case is impossible.*

(b) *If Q_α has one successor (K) : First, the case where A_α is irreducible is trivially impossible, since an AC-reflexion step of the clause $\neg(u \simeq v) \vee C'_1 \vee C'_2$ (C) produces the clause $C'_1\tau \vee C'_2\tau$, where τ is an AC-unifier of u and v. That new clause belongs to S_{j+1} and to Σ_K. So, a failure node should occur above K.*
The last case is that A_α is reducible by an atom $(g \simeq d)$ with $g \succ d$ (if $g =_{AC} d$, there should be a failure node above K).
We know the atom $A_{\beta'}$, corresponding to the literal $L''[l'\rho\sigma]$, is reducible by the atom $(l'\rho\sigma \simeq r'\rho\sigma)$, and $l'\rho\sigma \succ r'\rho\sigma$.
Since $Q_\alpha(L[d]) \neq Q_\alpha(L''[r'\rho\sigma])$, $Q_\alpha(A_\alpha[d]) \neq Q_\alpha(A_{\beta'}[r'\rho\sigma])$, abnd by Lemma 30, a failure node should cut the quasi-rightmost path, which implies that this case is impossible too.

2. *<u>If C' is stricly subsumed :</u> There is an index j such that : $C' \in S_j$ and a clause D' of S_j subsumes strictly C'. It means that there is a substitution ρ such that each literal of $D'\rho$ is AC-equal to a literal of C'.*

   So, $C = C_1 \vee C_2$ where $C_2 =_{AC} D'\rho\sigma$. Since $K(C_2) = F$, D' belongs to Σ_K, and, either $C_2 \prec_A C$ and C cannot belong to TG_K, or D' belongs to Π_K and, as D' subsumes strictly C', C' cannot be in Π'_K.

Since C' is never simplified or subsumed, it is persistent. □

Proposition 44 *The quasi-rightmost path Q_γ is AC-consistent.*

Proof : *The only point which can set a problem in the initial proof (Proposition 27) is the application of an extended paramodulation step between two clauses labeling failure nodes, which are right extensions of restrictions of the quasi-rightmost path.*

But, these failure nodes are extensions of AC-consistent interpretations. The proposition 43 ensures that the clauses C_1 and C_2 labeling them are persistent. So, the extended paramodulation step is always possible and the deduced clause C belongs to $\bigcap_{i\geq j} RP(S_i)$ and to Σ_{Q_γ}. The fairness of the derivation ensures that C is subsumed by a clause C' of $\bigcup_{i\geq 0} S_i$. So, Π'_{Q_γ} is not empty and a failure node should cut the quasi-rightmost path. □

Theorem 45 *Let S be an AC-unsatisfiable set of clauses. Then, every fair derivation from S generates the empty clause.*

Proof : *We will not describe the entire proof of this theorem, since all subcases are solved in the same way. So, let us detail proof of case 2.1.1 (Proposition 35), where the E-interpretation K, last node of the quasi-rightmost path of $MCT(\mathcal{S}^*)$, built as in the initial proof, admits two successors L and R which are failure nodes of $\mathcal{S}^*$. Let C_L and C_R be two clauses of $\mathcal{GS}$ respectively falsified by L and R. C'_L and C'_R denote the clauses of S^* corresponding to C_L and C_R.*

By the proof of proposition 35, we know there is a clause C' in $RP(\{C'_L, C'_R\})$ falsified by K, obtained by AC-resolution between C'_L and C'_R. By proposition 43, we can assume that C'_L and C'_R are persistent, i.e. : $C'_L, C'_R \in \bigcap_{i\geq j} S_i$. Hence, $C' \in \bigcap_{i\geq j} RP(S_i)$. The hypothesis of fairness of the derivation ensures that C' is subsumed by a clause D' of S^*. Therefore, K falsifies the clause D', and also cannot belong to $MCT(\mathcal{S}^*)$.

The proofs of all other subcases are similar, since we can follow the same reasoning, assuming that initial clauses are persistent, as C'_L and C'_R. □

7.3 Other Simplification Rules

A number of other simplification rules are compatible with AC-paramodulation. For instance, we can delete *tautologies* ; there are two kinds of tautologies : clauses that contain a literal AC-equal to an identity $(s \simeq s)$; and clauses that contain complementary literals, i.e. literals L and $\neg L'$, where L and L' are AC-equal.
Another important rule is the *clausal simplification* rule, which deletes all instances of a literal L in every clause of S, if $\neg L$ is a clause of S. We can also delete every instance of $\neg(x \simeq x)$ (it is a kind of trivial reflexion). A clausal simplification step may be viewed as a resolution step, followed by the deletion of one of the parent clauses.

Other reductions are possible, as *reductions by replacement*, i.e. to replace a clause $\neg(s \simeq t) \vee D$, where $s \succ t$, by the clause $\neg(s \simeq t) \vee D'$, where D' is equal to D, except that all subterms AC-equal to s have been replaced by t.

8 Implementation

The system of inference rules described in this paper has been implemented in a theorem-prover named `DATAC`. Another strategy is also available in `DATAC` : the positive ordered AC-paramodulation[3], which is based on the idea that any inference step uses at least one positive clause.

`DATAC` is written in CAML Light, a functional language of the ML family. It runs on SUN, HP and IBM PC Workstations. The AC-unification (resp. AC-matching) procedure is based on

[3]This system was presented at the First CCL Workshop, Val d'Ajol, France, 1992

Stickel's (resp. Hullot's) algorithm. In our implementation, we try to balance the complexity of AC-unification by applying a number of simplification rules for keeping the search space as small as possible. We have implemented a variant of the *associative path ordering* ([BP85]) for comparing terms and atoms.

`DATAC` is entirely automatic : after the user has entered a problem, specified by a set of first-order clauses, he gives a precedence between operators and predicates. Many parameters can be set before an execution. It is possible to limit the number of AC-unifiers computed, to limit the number and/or the size of generated clauses, to use different trace degrees, to enable or disable simplification rules. There is also a "step by step" mode where choices of clauses and inference rules are performed by the user. Our first experimentations on non trivial examples were encouraging.

The problem that we solve in the following was cited in [Koz91]. It is the proof of $(x^{**} \simeq x^*)$ in Kleene Algebra. The resolution uses the inference and simplification rules described in this paper. Kleene Algebra is represented by the first 13 clauses, and the negation of the theorem by the 14th. The operator $+$ is AC, and the precedence used is : $\cdot >_p + >_p {}^* >_p 1 >_p 0$.

>>> Initial set of clauses :

Clause 1 : $(x_1 + 0 \simeq x_1)$
Clause 2 : $(x_1 + x_1 \simeq x_1)$
Clause 3 : $((x_1 \cdot x_2) \cdot x_3 \simeq x_1 \cdot (x_2 \cdot x_3))$
Clause 4 : $(1 \cdot x_1 \simeq x_1)$
Clause 5 : $(x_1 \cdot 1 \simeq x_1)$
Clause 6 : $(x_1 \cdot (x_2 + x_3) \simeq (x_1 \cdot x_2) + (x_1 \cdot x_3))$
Clause 7 : $((x_1 + x_2) \cdot x_3 \simeq (x_1 \cdot x_3) + (x_2 \cdot x_3))$
Clause 8 : $(0 \cdot x_1 \simeq 0)$
Clause 9 : $(x_1 \cdot 0 \simeq 0)$
Clause 10 : $((x_1 \cdot x_1^*) + 1 \simeq x_1^*)$
Clause 11 : $((x_1^* \cdot x_1) + 1 \simeq x_1^*)$
Clause 12 : $\neg((x_1 \cdot x_2) + x_2 \simeq x_2) \ \vee \ ((x_1^* \cdot x_2) + x_2 \simeq x_2)$
Clause 13 : $\neg((x_1 \cdot x_2) + x_1 \simeq x_1) \ \vee \ ((x_1 \cdot x_2^*) + x_1 \simeq x_1)$
Clause 14 : $\neg(A^{**} \simeq A^*)$

>>> AC-Extended Paramodulation between 2 and 11
Clause 17 : $(1 + (x_1^* \cdot x_1) \simeq x_1^* + 1)$

>>> AC-Simplification from 11 into 17
Clause 17 : $(x_1^* + 1 \simeq x_1^*)$

>>> AC-Extended Paramodulation between 17 and 11
Clause 22 : $(x_1^* + (x_2^* \cdot x_2) \simeq x_2^* + x_1^*)$

>>> AC-Paramodulation from 17 into 6
Clause 50 : $((x_1 \cdot 1) + (x_1 \cdot x_2^*) \simeq x_1 \cdot x_2^*)$

>>> AC-Simplification from 5 into 50
Clause 50 : $(x_1 + (x_1 \cdot x_2^*) \simeq x_1 \cdot x_2^*)$

>>> AC-Paramodulation from 17 into 7
Clause 52 : $((1 \cdot x_2) + (x_1^* \cdot x_2) \simeq x_1^* \cdot x_2)$

>>> AC-Simplification from 4 into 52
Clause 52 : $(x_1 + (x_2^* \cdot x_1) \simeq x_2^* \cdot x_1)$

>>> AC-Simplification from 52 into 12
Clause 12 : $\neg((x_1 \cdot x_2) + x_2 \simeq x_2) \vee (x_1^* \cdot x_2 \simeq x_2)$

>>> AC-Simplification from 50 into 13
Clause 13 : $\neg((x_1 \cdot x_2) + x_1 \simeq x_1) \vee (x_1 \cdot x_2^* \simeq x_1)$

>>> AC-Paramodulation from 12 into 11
Clause 80 : $\neg((x_1 \cdot x_1) + x_1 \simeq x_1) \vee (x_1 + 1 \simeq x_1^*)$

>>> AC-Paramodulation from 50 into 80
Clause 214 : $\neg(x_1^* \cdot x_1^* \simeq x_1^*) \vee (x_1^* + 1 \simeq x_1^{**})$

>>> AC-Simplification from 17 into 214
Clause 214 : $\neg(x_1^* \cdot x_1^* \simeq x_1^*) \vee (x_1^{**} \simeq x_1^*)$

>>> AC-Paramodulation from 13 into 214
Clause 274 : $\neg((x_1^* \cdot x_1) + x_1^* \simeq x_1^*) \vee \neg(x_1^* \simeq x_1^*) \vee (x_1^{**} \simeq x_1^*)$

>>> Trivial AC-Reflexion in 274
Clause 274 : $\neg((x_1^* \cdot x_1) + x_1^* \simeq x_1^*) \vee (x_1^{**} \simeq x_1^*)$

>>> AC-Simplification from 22 into 274
Clause 274 : $\neg(x_1^* + x_1^* \simeq x_1^*) \vee (x_1^{**} \simeq x_1^*)$

>>> AC-Clausal Simplification in 274 thanks to 2
Clause 274 : $(x_1^{**} \simeq x_1^*)$

>>> AC-Clausal Simplification in 14 thanks to 274
Clause 14 : $\square$

>>> The initial system is inconsistent <<<

9 Further Works and Conclusion

We have designed a refutationally complete paramodulation-based strategy for associative-commutative deduction. It has been adapted from the ordered paramodulation strategy of [Pet83, HR91]. We have shown that AC-paramodulation is compatible with associative-commutative simplification. This is a fundamental issue for the efficiency of theorem-proving. Our strategy has been implemented in the system `DATAC` and experiments with non-trivial examples are encouraging. A similar treatment has been applied to another strategy, the positive strategy, and we think that our method could be applied to other theorem-proving strategies such as superposition [Rus89]. Moreover, some use of extended clauses could be avoided, as described in [BD89], where Bachmair and Dershowitz define a set of useful extended rules for a given rewriting system.

The idea of replacing axioms by ad-hoc mechanisms such as unification algorithm or inference rules can be further extended to other equational or non equational theories. In general the efficiency gain is noticeable, but this still needs to be carefully studied by experimentations.

References

[AHM89] S. Anantharaman, J. Hsiang, and J. Mzali. Sbreve2: A term rewriting laboratory with (AC-)unfailing completion. In N. Dershowitz, editor, *Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, volume 355 of *Lecture Notes in Computer Science*, pages 533–537. Springer-Verlag, April 1989.

[BCL87] A. Ben Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9(2):137–160, October 1987.

[BD89] L. Bachmair and N. Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2-3):173–202, October 1989.

[BG90] L. Bachmair and H. Ganzinger. On restrictions of ordered paramodulation with simplification. In M. E. Stickel, editor, *Proceedings 10th International Conference on Automated Deduction, Kaiserslautern (Germany)*, volume 449 of *Lecture Notes in Computer Science*, pages 427–441. Springer-Verlag, July 1990.

[BHK+88] H.-J. Bürckert, A. Herold, D. Kapur, J. Siekmann, M. E. Stickel, M. Tepp, and H. Zhang. Opening the AC-unification race. *Journal of Automated Reasoning*, 4(1):465–474, 1988.

[BP85] L. Bachmair and D. Plaisted. Associative path orderings. In *Proceedings 1st Conference on Rewriting Techniques and Applications, Dijon (France)*, volume 202 of *Lecture Notes in Computer Science*. Springer-Verlag, 1985.

[Bra75] D. Brand. Proving theorems with the modification method. *SIAM Journal of Computing*, 4:412–430, 1975.

[Bün91] R. Bündgen. Simulating Buchberger's algorithm by Knuth-Bendix completion. In R. V. Book, editor, *Proceedings 4th Conference on Rewriting Techniques and Applications, Como (Italy)*, volume 488 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, April 1991.

[Der82] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17:279–301, 1982.

[Dom91] E. Domenjoud. *Outils pour la déduction automatique dans les théories associatives-commutatives.* Thèse de Doctorat d'Université, Université de Nancy 1, September 1991.

[HR87] J. Hsiang and M. Rusinowitch. On word problem in equational theories. In T. Ottmann, editor, *Proceedings of 14th International Colloquium on Automata, Languages and Programming, Karlsruhe (Germany)*, volume 267 of *Lecture Notes in Computer Science*, pages 54–71. Springer-Verlag, 1987.

[HR91] J. Hsiang and M. Rusinowitch. Proving refutational completeness of theorem proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, July 1991.

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

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

[Koz91] D. Kozen. A completeness theorem for kleene algebras and the algebra of regular events. In *Proc. 6th IEEE symposium on Logic in Computer Science (LICS)*, pages 214–225. IEEE Computer Society Press, Los Alamitos, July 1991.

[KR87] E. Kounalis and M. Rusinowitch. On word problem in Horn logic. In J.-P. Jouannaud and S. Kaplan, editors, *Proceedings 1st International Workshop on Conditional Term Rewriting Systems, Orsay (France)*, volume 308 of *Lecture Notes in Computer Science*, pages 144–160. Springer-Verlag, July 1987. See also the extended version published in Journal of Symbolic Computation, 11(1 & 2), 1991.

[Lai89] M. Lai. On how to move mountains 'associatively and commutatively'. In N. Dershowitz, editor, *Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, volume 355 of *Lecture Notes in Computer Science*, pages 187–202. Springer-Verlag, April 1989.

[Lan79a] D. S. Lankford. Mechanical theorem proving in field theory. Technical report, Louisiana Tech. University, 1979.

[Lan79b] D. S. Lankford. On proving term rewriting systems are noetherian. Technical report, Louisiana Tech. University, Mathematics Dept., Ruston LA, 1979.

[LB77] D. S. Lankford and A. Ballantyne. Decision procedures for simple equational theories with associative commutative axioms: complete sets of associative commutative reductions. Technical report, Univ. of Texas at Austin, Dept. of Mathematics and Computer Science, 1977.

[Lov78] D. Loveland. *Automatic Theorem Proving.* Elsevier Science Publishers B. V. (North-Holland), 1978.

[NR91] P. Narendran and M. Rusinowitch. Any ground associative-commutative theory has a finite canonical system. In R. V. Book, editor, *Proceedings 4th Conference on Rewriting Techniques and Applications, Como (Italy)*. Springer-Verlag, 1991.

[Pau92] E. Paul. A general refutational completeness result for an inference procedure based on associative-commutative unification. *Journal of Symbolic Computation*, 14(6):577–618, 1992.

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

[Pet91] U. Petermann. Building in equational theories into the connection method. In P. Jorrand and J. Kelemen, editors, *Fundamental of Artificial Intelligence Research*, volume 535 of *Lecture Notes in Computer Science*, pages 156–169. Springer-Verlag, 1991.

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

[PP91] J. Pais and G. E. Peterson. Using forcing to prove completeness of resolution and paramodulation. *Journal of Symbolic Computation*, 11(1 & 2):3–19, 1991.

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

[Rus89] M. Rusinowitch. *Démonstration automatique-Techniques de réécriture*. InterEditions, 1989.

[RV91] M. Rusinowitch and L. Vigneron. Automated deduction with associative commutative operators. In P. Jorrand and J. Kelemen, editors, *Fundamental of Artificial Intelligence Research*, volume 535 of *Lecture Notes in Computer Science*, pages 185–199. Springer-Verlag, 1991.

[RV95] M. Rusinowitch and L. Vigneron. Automated Deduction with Associative-Commutative Operators. *Applicable Algebra in Engineering, Communication and Computation*, 6(1):23–56, January 1995.

[RW69] G. A. Robinson and L. T. Wos. Paramodulation and first-order theorem proving. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*, pages 135–150. Edinburgh University Press, 1969.

[Sti81] M. E. Stickel. A unification algorithm for associative-commutative functions. *Journal of the ACM*, 28:423–434, 1981.

[Sti84] M. E. Stickel. A case study of theorem proving by the Knuth-Bendix method: Discovering that $x^3 = x$ implies ring commutativity. In R. Shostak, editor, *Proceedings 7th International Conference on Automated Deduction, Napa Valley (Calif., USA)*, volume 170 of *Lecture Notes in Computer Science*, pages 248–258. Springer-Verlag, 1984.

[Wer92] U. Wertz. First-order theorem proving modulo equations. Technical Report MPI-I-92-216, Max Planck Institut für Informatik, April 1992.