AC-superposition with constraints: No AC-unifiers needed

Robert Nieuwenhuis and Albert Rubio*
Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain
E-mail: {roberto,rubio}@lsi.upc.es. Tel. 34-3-4017330

Abstract

We prove the completeness of (basic) deduction strategies with constrained clauses modulo associativity and commutativity (AC). Here each inference generates *one single conclusion* with an additional equality $s =_{\mathrm{AC}} t$ in its constraint (instead of one conclusion for each minimal AC-unifier, i.e. exponentially many). Furthermore, *computing AC-unifiers is not needed at all.* A clause $C \llbracket T \rrbracket$ is redundant if the constraint T is not AC-unifiable. If C is the empty clause this has to be decided to know whether $C \llbracket T \rrbracket$ denotes an inconsistency. In all other cases any sound method to detect unsatisfiable constraints can be used.

1 Introduction

Some fundamental ideas on applying symbolic constraints to theorem proving were given in [KKR90], where a constrained clause is a shorthand for its (infinite) set of ground instances satisfying the constraint. In a constrained equation $f(x) \simeq a \llbracket x = g(y) \rrbracket$, the equality '=' of the constraint is usually interpreted in $\mathcal{T}(\mathcal{F})$ (syntactic equality), or in some quotient algebra $\mathcal{T}(\mathcal{F})/\equiv_E$ where E is an equational theory. The methods in [KKR90] require to *propagate* parts of the constraints to the clause. For example, if '=' is syntactic equality, by such a propagation step the previous equation can be replaced by the logically equivalent one $f(g(y)) \simeq a \llbracket true \rrbracket$.

Avoiding propagation is essential for exploiting the constraints. By doing so for (syntactic) equality constrained clauses, in [NR92a] the completeness of *basic superposition* is proved (independently also in [BGLS92]), and by adding *ordering constraints* the search space is further reduced by *inheriting* the ordering restrictions of the inference rules ([NR92b]). Constrained (equational) superposition can e.g. be expressed like:

$$\frac{s' \simeq t' \llbracket T' \rrbracket \qquad s \simeq t \llbracket T \rrbracket}{s[t']_p \simeq t \llbracket T' \wedge T \wedge s' \succ t' \wedge s \succ t \wedge s|_p = s \rrbracket} \quad \text{where } s|_p \notin \mathcal{V}ars(s)$$

In this (elegant and powerful) representation for ordered inference rules, information from the meta-level (the ordering and unifiability restrictions) is kept and inherited to restrict future inferences: clauses with unsatisfiable constraints are tautologies, hence redundant. These inference rules are *basic* (i.e. no inferences are needed on subterms introduced by the unifiers of previous inferences) because the accumulated unification problems are kept in the equality constraints (no propagation is needed) and future superpositions can take place only on non-variable subterms of the clause part.

*This work has been partially supported by the Esprit Working Group CCL, ref. 6028

Here we use these techniques for deduction modulo the theory of associativity and commutativity (AC), applying in our completeness proofs an essential ingredient which we recently gave for this purpose in [RN93]: an AC-compatible simplification ordering that is total on AC-distinct ground terms. This ordering is also defined on terms with variables, which makes it applicable in practice for checking the —non-ground— ordering restrictions (the first and only —as far as we know— other such ordering of [NR91] is defined only on ground terms). Our completeness proofs are based on the *model generation* framework with its *abstract redundancy notions* for detecting redundant clauses and inferences during the theorem proving process, defined by Bachmair and Ganzinger in [BG91]. Here we have adapted these techniques to AC-deduction, similarly to Wertz's work ([Wer92]), although he deals with explicit *extended* clauses, while we simulate them by means of specific AC-inference rules, like in [RV93]. We believe that our treatment of the AC-case is interesting in itself (apart from the constraints) because of its simplicity. Albert Rubio's PhD. thesis [Rub94] contains a very complete analysis of all these constrained deduction methods (also covering the AC case).

We prove the refutation completeness of a superposition-based inference system, where each inference has *one single conclusion* with an additional equality $s =_{\mathrm{AC}} t$ in its constraint, instead of one conclusion for each minimal AC-unifier (double exponentially many: e.g. $x + x + x$ and $y_1 + y_2 + y_3 + y_4$ have more than a million minimal unifiers [Dom92]). This also eliminates the need of computing AC-unifiers in AC-deduction (or completion) methods [PS81, JK86, BD89] which have motivated a huge amount of research on computing complete sets of AC-unifiers (see [BS93] for a recent survey). A clause C with an AC-equality constraint T of the form $s_1 =_{\mathrm{AC}} t_1 \wedge \ldots \wedge s_n =_{\mathrm{AC}} t_n$ can be proved redundant by means of efficient incomplete methods detecting cases of unsatisfiability of T. If C is the empty clause one can decide the AC-unifiability of T (which is NP-complete, cf. [KN92]) to know whether an inconsistency has been derived or not.

The first results on (almost basic) constrained deduction modulo AC were reported by Laurent Vigneron. In a recent version of his work [Vig94] he also avoids the computation of AC-unifiers (by applying our notion of irreducibility, defin. 3.3) and defines several additional restrictions. His proofs are completely different from ours and based on transfinite semantic trees as in [RV93]. He also reports on an implementation of these methods. Advantages of our application of the model generation method wrt. semantic trees are simplicity: our full proofs can be given in a very reasonable amount of space (this paper), and also that the known extensions to the model generation method for constrained paramodulation like *redex orderings* and *variable abstraction* [BGLS92] can be smoothly incorporated here. Similarly, the *abstract* redundancy notions express very sharp bounds on *concrete* redundancy methods (like the ones given by Vigneron which indeed fit into our abstract ones).

In section 3 of this paper we give a superposition-based inference system, which is proved refutationally complete in section 4. Section 5 is on the compatibility with simplification and deletion methods, and section 6 is on Knuth-Bendix-like completion of (conditional) equations modulo AC.

2 Inference rules

Definition 2.1 *The following inference system is called* BOAC *(for basic, ordering constrained and AC). It is* strict superposition-based, *i.e. the ordering constraints OC added in all rules always encode that the equations $s \simeq t$ and $s' \simeq t'$ are maximal wrt. $\succ$ in the premise to which they belong (strictly maximal if they belong to the succedent, except the factoring rule, where only $s \simeq t$ is maximal), with $s \succ t$ and $s' \succ t'$ (except in equality resolution).*

In the AC rules, where x and x' are new variables, the term s' can be restricted to be headed by the AC-symbol f. This can be expressed in the constraint language and added to the constraint. As usual, AC-top superposition is only needed if s and s' are headed by f and share some top-level subterm (not headed with f) but x and x' do not (some restrictions implied by this condition can be formulated in the constraint language).

Of course, the superposition inferences are needed only if $s|_p$ is non-variable, and AC-superposition is needed only if moreover $s|_p$ (which has an f as top symbol) is not immediately below another f. Some examples are given at the end of next section.

1. *strict superposition right:*

$$\frac{\Gamma' \rightarrow \Delta', s' \simeq t' \; [\![\, T' \,]\!] \qquad \Gamma \rightarrow \Delta, s \simeq t \; [\![\, T \,]\!]}{\Gamma', \Gamma \rightarrow \Delta', \Delta, s[t']_p \simeq t \; [\![\, T' \wedge T \wedge OC \wedge \; s|_p =_{\mathrm{AC}} s' \,]\!]}$$

2. *strict superposition left:*

$$\frac{\Gamma' \rightarrow \Delta', s' \simeq t' \; [\![\, T' \,]\!] \qquad \Gamma, s \simeq t \rightarrow \Delta \; [\![\, T \,]\!]}{\Gamma', \Gamma, s[t']_p \simeq t \rightarrow \Delta', \Delta \; [\![\, T' \wedge T \wedge OC \wedge \; s|_p =_{\mathrm{AC}} s' \,]\!]}$$

3. *equality resolution:*

$$\frac{\Gamma, s \simeq t \rightarrow \Delta \; [\![\, T \,]\!]}{\Gamma \rightarrow \Delta \; [\![\, T \wedge OC \wedge \; s =_{\mathrm{AC}} t \,]\!]}$$

4. *factoring:*

$$\frac{\Gamma \rightarrow \Delta, s' \simeq t', s \simeq t \; [\![\, T \,]\!]}{\Gamma, t \simeq t' \rightarrow \Delta, s \simeq t \; [\![\, T \wedge OC \wedge \; s =_{\mathrm{AC}} s' \,]\!]}$$

5. *AC-strict superposition right:*

$$\frac{\Gamma' \rightarrow \Delta', s' \simeq t' \; [\![\, T' \,]\!] \qquad \Gamma \rightarrow \Delta, s \simeq t \; [\![\, T \,]\!]}{\Gamma', \Gamma \rightarrow \Delta', \Delta, s[f(t', x)]_p \simeq t \; [\![\, T' \wedge T \wedge OC \wedge \; s|_p =_{\mathrm{AC}} f(s', x) \,]\!]}$$

6. *AC-strict superposition left:*

$$\frac{\Gamma' \rightarrow \Delta', s' \simeq t' \; [\![\, T' \,]\!] \qquad \Gamma, s \simeq t \rightarrow \Delta \; [\![\, T \,]\!]}{\Gamma', \Gamma, s[f(t', x)]_p \simeq t \rightarrow \Delta', \Delta \; [\![\, T' \wedge T \wedge OC \wedge \; s|_p =_{\mathrm{AC}} f(s', x) \,]\!]}$$

7. *AC-top-superposition:*

$$\frac{\Gamma' \rightarrow \Delta', s' \simeq t' \; [\![\, T' \,]\!] \qquad \Gamma \rightarrow \Delta, s \simeq t \; [\![\, T \,]\!]}{\Gamma', \Gamma \rightarrow \Delta', \Delta, f(t', x') \simeq f(t, x) \; [\![\, T' \wedge T \wedge OC \wedge \; f(s', x') =_{\mathrm{AC}} f(s, x) \,]\!]}$$

3 Refutation completeness

Definition 3.1 *Let $C \llbracket T \rrbracket$ be a constrained clause and let σ be a ground substitution such that $T\sigma$ is true. Then $C\sigma$ is called a ground* instance *with σ of $C \llbracket T \rrbracket$.*

If moreover $C\sigma$ is of the form $\Gamma \rightarrow \Delta, t \simeq t'$ where $top(t)$ is an AC symbol f, and $t \simeq t'$ is the strictly maximal equation in $C\sigma$ with $t \succ t'$, then for every ground term v, the clause $\Gamma \rightarrow \Delta, f(t, v) \simeq f(t', v)$ is also a ground instance of $C \llbracket T \rrbracket$, called an extended instance *(with the* context *v) of $C\sigma$.*

Definition 3.2 *If, for a given ground instance $C\sigma$ of a clause C, a variable x only appears in equations $x \simeq t$ of the succedent of C with $x\sigma \succ t\sigma$ then x is called a succedent-top variable of $C\sigma$, denoted $x \in stvars(C\sigma)$.*

Definition 3.3 *Let R be a set of ground rewrite rules, let $C\sigma$ be a ground instance of a clause C and let x be a variable in $\mathcal{V}ars(C)$. Then x is said to be variable irreducible in $C\sigma$ wrt. R if,*

1. *$x\sigma$ is irreducible wrt. R, or*

2. *$x \in stvars(C\sigma)$ and $x\sigma$ is irreducible wrt. all rules $l \rightarrow r \in R$ s.t. $x\sigma \simeq t\sigma \succ l \simeq r$ for all $x \simeq t$ in C, or*

3. *x only occurs in C immediately below some AC-symbol f and all subterms t of $x\sigma$ with $top(t) \neq f$ are irreducible wrt. R.*

If this property holds for all x in $\mathcal{V}ars(C)$ then $C\sigma$ is variable irreducible[1] *wrt. R.*

Definition 3.4 *Let AC_C denote the set of ground instances $s \simeq s'$ of equations in AC with $C \succ s \simeq s'$ and s and s' irreducible wrt. R_C, and let I_C denote the interpretation $(R_C \cup AC_C)^*$.*

If C is a ground instance of a clause in S of the form $\Gamma \rightarrow \Delta, s \simeq t$ where $s \simeq t$ is strictly maximal (wrt. $\succ$) in C with $s \succ t$ and $I_C \not\models \Gamma \rightarrow \Delta$ then

1. *if C is a non-extended instance that is variable irreducible wrt. R_C and all u with $u =_{\mathrm{AC}} s$ are irreducible by R_C, then C generates all rules $u \Rightarrow t$ with $u =_{\mathrm{AC}} s$.*

2. *if C is an extended instance of some D that has generated some rule, then for each u with $u =_{\mathrm{AC}} s$ and u irreducible by R_C, C generates a rule $u \Rightarrow t$.*

Finally, we define $R_S = \bigcup R_C$, $AC_S = \bigcup AC_C$, and $I_S = (R_S \cup AC_S)^*$.

Lemma 3.5 *Let S be a set of constrained clauses, and C and D instances of clauses in S with $C \succ D$. Then $R_C \supseteq R_D$, $AC_C \supseteq AC_D$, and $I_C \supseteq I_D$.*

[1] Here point 2. is based on the irreducibility notion of [BGLS92], and point 3. is the crucial trick in our proof for lifting in the AC-case and thus avoiding the computation of AC-unifiers, cf. example 3.13. Vigneron applies our idea in [Vig94] and extends it by allowing point 3 only if x is a new variable of AC-superposition, which permits to impose some more restrictions on the inferences (this also works for the proofs given here).

Proof *The first point, $R_C \supseteq R_D$, holds by definition. For $AC_C \supseteq AC_D$: suppose some $s \simeq s' \in AC_D$ and $s \simeq s' \notin AC_C$. Then there is some instance D' producing a rule $l \Rightarrow r$ that reduces s or s', with $s \simeq s' \in AC_{D'}$. But the equation $l \simeq r$ is strictly maximal in D' with $l \succ r$ and since $l \Rightarrow r$ reduces $s \simeq s'$, we have $s \succeq l$ and $s \succ r$, i.e. $s \simeq s' \succ l \simeq r$ and therefore $s \simeq s' \succ D'$, which contradicts $s \simeq s' \in AC_{D'}$. From the two previous points follows $I_C \supseteq I_D$.* □

Lemma 3.6 *Let S be a set of constrained clauses.*

1. *If instances C and D with $C \succ D$ generate resp. $l \Rightarrow r$ and $l' \Rightarrow r'$ then $l \succ l'$.*
2. *For all $s_1 \simeq s_2$ in AC_S, the terms s_1 and s_2 are irreducible wrt. R_S.*
3. *For all $l \Rightarrow r$ and $l' \Rightarrow r'$ in R_S, if $l =_{\mathrm{AC}} l'$ then r and r' are the same term.*
4. *There are no overlaps between left hand sides of rules of R_S.*
5. *If $u =_{\mathrm{AC}} v$ for ground terms u and v, then u is reducible by R_S iff v is.*

Lemma 3.7 *Let S be a set of constrained clauses, and let s and t be ground terms. Then $I_S \models s \simeq t$ implies $s \rightarrow^*_{R_S} s' =_{\mathrm{AC}_S} t' \leftarrow^*_{R_S} t$ for some s' and t'.*

Proof *If $I_S \models s \simeq t$ then there is a proof $s \leftrightarrow^*_{R_S \cup AC_S} t$, whose critical peaks can be overlaps of the form $t_1 \leftarrow_{R_S} t_2 \rightarrow_{R_S} t_3$, or of the form $t_1 \leftrightarrow_{AC_S} t_2 \rightarrow_{R_S} t_3$. The first situation cannot happen (by point 4. of the previous lemma). The other kind of overlap must be of the form $u[l[s_1]] \leftrightarrow_{AC_S} u[l[s_2]] \rightarrow_{R_S} u[r]$ (by point 2.). Since there is a rule $l[s_2] \Rightarrow r$, and $l[s_1] =_{\mathrm{AC}} l[s_2]$, the term $l[s_1]$ is also reducible by R_S (point 5.), and moreover, it must be reducible at topmost position, because otherwise the corresponding AC-equal subterm in $l[s_2]$ would also be reducible. Now if $l[s_2]$ is reducible by some rule $l[s_2] \Rightarrow r'$, then $r' = r$ (point 3.). The fact that all overlaps $t_1 \leftrightarrow_{AC_S} t_2 \rightarrow_{R_S} t_3$ can be directly replaced by $t_1 \rightarrow_{R_S} t_3$ implies (by induction on the number of such AC-steps in $s \leftrightarrow^*_{R_S \cup AC_S} t$) that in fact $s \rightarrow^*_{R_S} s' =_{\mathrm{AC}_S} t' \leftarrow^*_{R_S} t$.* □

Lemma 3.8 *If $\Gamma\sigma \rightarrow \Delta\sigma, s\sigma \simeq t\sigma$ is an instance C of S that generates rules $u \Rightarrow t\sigma$ for all $u =_{\mathrm{AC}} s\sigma$ then $I_S \not\models \Gamma\sigma \rightarrow \Delta\sigma$.*

Proof *Since C generates rules, $I_C \not\models \Gamma\sigma \rightarrow \Delta\sigma$. If $I_C \supseteq \Gamma\sigma$ then also $I_S \supseteq \Gamma\sigma$. It remains to be shown that $I_S \cap \Delta\sigma = \emptyset$.*

We know $I_C \cap \Delta\sigma = \emptyset$, and the rules generated by instances bigger than C have left hand sides bigger than u, so they cannot contribute to rewrite proofs of $\Delta\sigma$, and the only rules in $R_S \setminus R_C$ that could be used are the $u \Rightarrow t\sigma$, if there is some equation $s'\sigma \simeq t'\sigma$ in $\Delta\sigma$ such that $s'\sigma$ is one of the u's and $I_C \models t\sigma \simeq t'\sigma$. But then the following inference by factoring can be made:

$$\frac{\Gamma \rightarrow \Delta, s' \simeq t', s \simeq t \;[\![\, T \,]\!]}{\Gamma, t' \simeq t \rightarrow \Delta, s \simeq t \;[\![\, T \wedge OC \wedge\ s =_{\mathrm{AC}} s' \,]\!]}$$

Its conclusion has a ground instance D smaller than C of the form $\Gamma\sigma, t'\sigma \simeq t\sigma \rightarrow \Delta\sigma, s\sigma \simeq t\sigma$ where $I_D \not\models D$ and which generates the rules $u \Rightarrow t\sigma$ for all $u =_{\mathrm{AC}} s\sigma$, contradicting the fact that they have been generated by C. □

Lemma 3.9 *Let S be a set of constrained clauses. If a ground instance $C\sigma$ of a clause $C \llbracket T \rrbracket$ in S, with C of the form $\Gamma \rightarrow \Delta, s \simeq t$ generates rules $u \Rightarrow t\sigma$ with $u =_{\mathrm{AC}} s\sigma$ then $C\sigma$ is variable irreducible wrt. R_S.*

Proof *We know $C\sigma$ is variable irreducible wrt. $R_{C\sigma}$, since it has generated a rule, so we only have to show the variable irreducibility of x in $\mathcal{V}ars(C)$ wrt. rules $l \Rightarrow r$ in $R_S \setminus R_{C\sigma}$. Now if $s\sigma \succ x\sigma$ then no such rule $l \Rightarrow r$ reduces $x\sigma$, since $l \succeq s\sigma \succ x\sigma$. Otherwise, since $s\sigma$ is the maximal term in $C\sigma$, we have $x\sigma =_{\mathrm{AC}} s\sigma$ and only some rule $u \Rightarrow t\sigma$ can reduce $x\sigma$. But then $x \in stvars(C\sigma)$ and $u \simeq t\sigma \succeq x\sigma \simeq t'\sigma$ for every $x \simeq t'$ in C, i.e. x is variable irreducible in $C\sigma$ wrt. R_S.* $\square$

Definition 3.10 *Let S be a set of constrained clauses. We denote by $irred(S)$ the set of non-extended ground instances of clauses in S that are variable irreducible wrt. R_S.*

Lemma 3.11 *Let S be a set of constrained clauses not containing the empty clause and closed under* BOAC*. Then $I_S \models irred(S) \cup AC$.*

Proof *We will derive a contradiction from the existence of a minimal (wrt. $\succ$) instance C in $irred(S) \cup AC$ such that $I_S \not\models C$.*

If C is an instance $f(u_1, u_2) \simeq f(u_2, u_1)$ of the commutativity axiom $f(x, y) \simeq f(y, x)$, then since $C \notin AC_S$, one of $f(u_1, u_2)$ and $f(u_2, u_1)$ is reducible by R_S and therefore both of them, since they are AC-equal. If u_1 or u_2 is reducible, e.g. u_1 rewrites into some u, then $I_S \not\models f(u, u_2) \simeq f(u_2, u)$ which is a smaller instance than C of AC, contradicting the minimality of C. (Note that this smaller instance really exists since the AC-axioms do not have constraints). If u_1 and u_2 are irreducible, then let wlog. $f(u_1, u_2)$ be reducible by a rule $f(u_1, u_2) \Rightarrow r$ in R_S. But then, since u_1 and u_2 are irreducible, the same instance that generates $f(u_1, u_2) \Rightarrow r$ also produces $f(u_2, u_1) \Rightarrow r$, contradicting $I_S \not\models f(u_1, u_2) \simeq f(u_2, u_1)$.

If C is an instance $f(f(u_1, u_2), u_3) \simeq f(u_1, f(u_2, u_3))$ of the associativity axiom, then since $C \notin AC_S$, one of $f(f(u_1, u_2), u_3)$ and $f(u_1, f(u_2, u_3))$ is reducible by R_S and therefore both of them, since they are AC-equal. If u_1, u_2 or u_3 is reducible, then a contradiction is obtained as in the previous case for commutativity. If one of $f(f(u_1, u_2), u_3)$ and $f(u_1, f(u_2, u_3))$ is reducible at topmost position by a rule $l \Rightarrow r$ generated by the non-extended instance then also the other one is reducible at the topmost position by a rule with the same right hand side, contradicting $I_S \not\models f(f(u_1, u_2), u_3) \simeq f(u_1, f(u_2, u_3))$.

Otherwise, let the rule reducing $f(f(u_1, u_2), u_3)$ be $l \Rightarrow r$ with $l =_{\mathrm{AC}} s\sigma$ and $r = t\sigma$ or $l =_{\mathrm{AC}} f(s\sigma, v)$ and $r = f(t\sigma, v)$ generated by an instance with σ (possibly extended with the context v) of a clause D of the form $\Gamma \rightarrow \Delta, s \simeq t$. Then $f(f(u_1, u_2), u_3) =_{\mathrm{AC}} f(s\sigma, v_1)$ for some v_1, where v_1 is, respectively, v (topmost reduction with extended rule), or u_3 (reduction of $f(u_1, u_2)$ with non-extended rule), or $f(u_3, v)$ (reduction of $f(u_1, u_2)$ with extended rule), and $f(f(u_1, u_2), u_3)$ is rewritten into some s_1 with $s_1 =_{\mathrm{AC}_{\prec C}} f(t\sigma, v_1)$ (we use $\mathrm{AC}_{\prec C}$ to denote the instances of AC that are smaller wrt. $\succ$ than C; note that $I_S \models \mathrm{AC}_{\prec C}$ by minimality of C).

Similarly, let the rule reducing $f(u_1, f(u_2, u_3))$ be $l' \Rightarrow r'$ with $l' =_{\mathrm{AC}} s'\sigma$ and $r' = t'\sigma$ or $l' =_{\mathrm{AC}} f(s'\sigma, v')$ and $r' = f(t'\sigma, v')$ generated by an instance with σ (possibly extended *with the context v') of a clause D' of the form $\Gamma' \to \Delta', s' \simeq t'$ (we can use the same σ since D and D' do not share variables). Then $f(u_1, f(u_2, u_3)) =_{\mathrm{AC}} f(s'\sigma, v_2)$ for some v_2, where v_2 is, respectively, v' (topmost reduction with extended rule), or u_1 (non-topmost reduction with non-extended rule), or $f(u_1, v')$ (non-topmost reduction with extended rule), and $f(u_1, f(u_2, u_3))$ is rewritten into some s_2 with $s_2 =_{\mathrm{AC}_{\prec C}} f(t'\sigma, v_2)$.*

Now there is an inference by AC-top-superposition between D' and D:

$$\frac{\Gamma' \to \Delta', s' \simeq t' \llbracket T' \rrbracket \qquad \Gamma \to \Delta, s \simeq t \llbracket T \rrbracket}{\Gamma', \Gamma \to \Delta', \Delta, f(t', x') \simeq f(t, x) \llbracket T' \wedge T \wedge OC \wedge \ f(s', x') =_{\mathrm{AC}} f(s, x) \rrbracket}$$

whose conclusion D_1 has an instance with a ground substitution θ defined by $\theta = \sigma \cup \{x \mapsto v_1, x' \mapsto v_2\}$.

$D_1\theta$ is variable irreducible wrt. R_S since $D'\theta$ and $D\theta$ are variable irreducible wrt. R_S (by lemma 3.9) and for every variable $y \in \mathcal{V}ars(D_1)$, either $y\theta$ is irreducible wrt. R_S or we have:

1. *$y \in stvars(D\theta)$ or $y \in stvars(D'\theta)$ in some $y \simeq t_y$ Then also $y \in stvars(D_1\theta)$ in the same $y \simeq t_y$, and therefore y is variable irreducible in $D_1\theta$ wrt. R_S.*

2. *y only occurs in D or D' immediately below some AC-symbol. Then y also only occurs in D_1 immediately below the same AC-symbol and therefore it is variable irreducible in $D_1\theta$ wrt. R_S.*

3. *y is one of the new variables, x or x' (wlog. x). Since $f(f(u_1, u_2), u_3) =_{\mathrm{AC}} f(s\sigma, v_1)$, for every subterm u at a maximal position of v_1 with $top(u) \neq f$ there is an u' with $u =_{\mathrm{AC}} u'$ in u_1, u_2 or u_3, which implies that such u are irreducible wrt. R_S, and since x occurs only in D_1 immediately below the AC-symbol f, it follows that x is variable irreducible in $D_1\theta$ wrt. R_S.*

$D_1\theta$ is an existing instance smaller than C, and moreover variable irreducible, so from the minimality of C we have $I_S \models D_1\theta$. We know $I_S \not\models \Gamma'\sigma, \Gamma\sigma \to \Delta'\sigma, \Delta\sigma$, so it must be the case that $I_S \models f(t', x')\theta \simeq f(t, x)\theta$, i.e. $I_S \models f(t'\sigma, v_2) \simeq f(t\sigma, v_1)$, but then $f(f(u_1, u_2), u_3) \to_{R_S} s_1 =_{\mathrm{AC}_{\prec C}} f(t\sigma, v_1)$ and $f(u_1, f(u_2, u_3)) \to_{R_S} s_2 =_{\mathrm{AC}_{\prec C}} f(t'\sigma, v_2)$ contradict $I_S \not\models C$, which completes the proof for the case where C is an instance of associativity.

If C is an instance with σ of a clause $D \llbracket T \rrbracket$ in S, then there are several cases to be analyzed, depending on the maximal equation in C:

1. *C has two maximal equations in its succedent.*

2. *C has one strictly maximal equation in its succedent.*

3. *C has a maximal equation $u \simeq u'$ in its antecedent, with $u =_{\mathrm{AC}} u'$.*

4. *C has a maximal equation $u \simeq u'$ in its antecedent, with $u \succ u'$.*

1. *C has two maximal equations in its succedent, i.e. D is $\Gamma \to \Delta, s' \simeq t', s \simeq t$, and there are two AC-equal maximal equations $s'\sigma \simeq t'\sigma$ and $s\sigma \simeq t\sigma$, that is, we have $s\sigma =_{\mathrm{AC}} s'\sigma \succ t\sigma =_{\mathrm{AC}} t'\sigma$, (note that $s\sigma \neq_{\mathrm{AC}} t\sigma$, since otherwise in fact $s\sigma =_{\mathrm{AC}_{\prec C}} t\sigma$, which, since $I_S \models \mathrm{AC}_{\prec C}$, would contradict $I_S \not\models C$). But then the following inference by factoring can be made:*

$$\frac{\Gamma \to \Delta, s' \simeq t', s \simeq t \,[\![\, T \,]\!]}{\Gamma, t' \simeq t \to \Delta, s \simeq t \,[\![\, T \wedge OC \wedge\ s =_{\mathrm{AC}} s' \,]\!]}$$

Its conclusion D_1 has a ground instance $D_1\sigma$ of the form $\Gamma\sigma, t'\sigma \simeq t\sigma \to \Delta\sigma, s\sigma \simeq t\sigma$ which is not satisfied by I_S, is smaller than C, and variable irreducible (as above) wrt. R_S. This contradicts the minimality of C.

2. *Let $D\sigma$ be a clause $\Gamma\sigma \to \Delta\sigma, s\sigma \simeq t\sigma$, with a strictly maximal equation $s\sigma \simeq t\sigma$, and $s\sigma \succ t\sigma$ (we have $s\sigma \neq_{\mathrm{AC}} t\sigma$ as in case 1, except if C consists only of the equation $s\sigma \simeq t\sigma$, but then $s\sigma =_{\mathrm{AC}} t\sigma$ would follow from $\mathrm{AC}_{\prec C}$ plus instances of AC that are AC-equal to C, which are shown true in I_S by the first part of this proof).*

Since $I_S \not\models C$, the instance C has not generated the rule $s\sigma \Rightarrow t\sigma$. This must be because some u with $s\sigma =_{\mathrm{AC}} u$ is reducible by R_C with a rule generated by an instance C' smaller than C.

Let this rule be the one that reduces u at a position p with $top(u|_p) = f$ (for some AC or non-AC symbol f) where p is innermost in the following sense: no other rule reduces u in a position p' below $\mathrm{maxpos}(u,p)$ with $top(u|_{p'}) \neq f$. Note that such a rule always exists. Now there are two main subcases **2.1** *and* **2.2**, *depending on whether the rule has been generated by an extended instance or by a non-extended instance.*

2.1 *Let C' be a non-extended instance $D'\sigma$ of some clause $D' \,[\![\, T' \,]\!]$ (we can use the same σ since D and D' do not share variables) with D' of the form $\Gamma' \to \Delta', s' \simeq t'$, where the rule is $u' \Rightarrow t'\sigma$ for some u' with $s'\sigma =_{\mathrm{AC}} u' = u|_p$.*

2.1.1 *If $\mathrm{maxpos}(u,p) = p$ then $u' =_{\mathrm{AC}} s\sigma|_{p'}$ for the corresponding maximal position p' in $s\sigma$ and therefore $s'\sigma =_{\mathrm{AC}} s\sigma|_{p'}$. Then $s\sigma|_{p'}$ cannot be below a variable, as $D\sigma$ is variable irreducible wrt. R_S: if s were in $stvars(D\sigma)$, then it would be irreducible wrt. rules smaller than $s\sigma \simeq t\sigma$, which is not the case; case 3. of definition 3.3 does not apply either, since $\mathrm{maxpos}(s\sigma, p') = p'$. Then the inference by strict superposition right*

$$\frac{\Gamma' \to \Delta', s' \simeq t' \,[\![\, T' \,]\!] \qquad \Gamma \to \Delta, s \simeq t \,[\![\, T \,]\!]}{\Gamma', \Gamma \to \Delta', \Delta, s[t']_{p'} \simeq t \,[\![\, T' \wedge T \wedge OC \wedge\ s|_{p'} =_{\mathrm{AC}} s' \,]\!]}$$

can be made, and, since all the conditions for its application hold, its conclusion D_1 has a ground instance $D_1\sigma$ of the form $\Gamma'\sigma, \Gamma\sigma \to \Delta'\sigma, \Delta\sigma, s\sigma[t'\sigma]_{p'} \simeq t\sigma$ that is not satisfied by I_S, is smaller than C, and (as above) variable irreducible wrt. R_S. This contradicts the minimality of C.

2.1.2 *If $p_1 = \mathrm{maxpos}(u,p) \neq p$ then $u|_{p_1} =_{\mathrm{AC}} s\sigma|_{p'}$ for the corresponding maximal position p' in $s\sigma$ and $u|_{p_1} =_{\mathrm{AC}} f(u', v) =_{\mathrm{AC}} f(s'\sigma, v)$ for some v, and therefore $s\sigma|_{p'} =_{\mathrm{AC}} f(s'\sigma, v)$. Then $s\sigma|_{p'}$ cannot be a variable, as before, since again $\mathrm{maxpos}(s\sigma, p') = p'$. Then the inference by AC-strict superposition right*

$$\frac{\Gamma' \to \Delta', s' \simeq t' \,[\![\, T' \,]\!] \qquad \Gamma \to \Delta, s \simeq t \,[\![\, T \,]\!]}{\Gamma', \Gamma \to \Delta', \Delta, s[f(t', x)]_{p'} \simeq t \,[\![\, T' \wedge T \wedge OC \wedge\ s|_{p'} =_{\mathrm{AC}} f(s', x) \,]\!]}$$

can be made, and, since all the conditions for its application hold, its conclusion a ground instance with σ of the form $\Gamma'\sigma, \Gamma\sigma \rightarrow \Delta'\sigma, \Delta\sigma, s\sigma[f(t'\sigma, v)]_{p'} \simeq t\sigma$ that is not satisfied by I_S, is smaller than C, and variable irreducible wrt. R_S (as above, and taking into account that all subterms u of $s\sigma|_{p'}$ with $top(u) \neq f$ are irreducible. wrt. R_S, since we have considered an innermost reduction in this sense). This again contradicts the minimality of C.

2.2 *The other subcase is that u is reducible by R_C with a rule generated by an extended instance C' smaller than C. Let C' be an extended instance of some clause D' $[\![T']\!]$ of the form $\Gamma' \rightarrow \Delta', s' \simeq t'$, where $top(s'\sigma)$ is the AC-symbol f and the rule is $u' \Rightarrow f(t'\sigma, v)$ for some u' with $u' =_{\mathrm{AC}} f(s'\sigma, v)$ and $u|_p = u'$.*

Let p_1 be $\mathrm{maxpos}(u, p)$. Then $u|_{p_1} =_{\mathrm{AC}} s\sigma|_{p'}$ for the corresponding position p' in $s\sigma$ and $u|_{p_1} =_{\mathrm{AC}} f(u', v') =_{\mathrm{AC}} f(s'\sigma, v')$ for some v', which is v if $p_1 = p$, and $f(v, v'')$ otherwise, for some v''. Therefore $s\sigma|_{p'} =_{\mathrm{AC}} f(s'\sigma, v')$ for some p' and v'. Then $s\sigma|_{p'}$ cannot be a variable, as before, since again $\mathrm{maxpos}(s\sigma, p') = p'$, another inference by AC-strict superposition right can be made, whose conclusion, as above, has an instance that contradicts the minimality of C.

3. *C has a maximal equation in its antecedent whose members are AC-equal, i.e. D is $\Gamma, s \simeq t \rightarrow \Delta$, and $s\sigma =_{\mathrm{AC}} t\sigma$. Then the following inference by equality resolution can be made:*

$$\frac{\Gamma, s \simeq t \rightarrow \Delta \;[\![T]\!]}{\Gamma \rightarrow \Delta \;[\![T \wedge OC \wedge \; s =_{\mathrm{AC}} t]\!]}$$

Its conclusion has a ground instance smaller than C of the form $\Gamma\sigma \rightarrow \Delta\sigma$ which is not satisfied by I_S, is smaller than C, and variable irreducible wrt. R_S, which contradicts the minimality of C.

4. *$D\sigma$ is $\Gamma\sigma, s\sigma \simeq t\sigma \rightarrow \Delta\sigma$, with a maximal equation $s\sigma \simeq t\sigma$, and $s\sigma \succ t\sigma$.*

Since $I_S \not\models C$, we have $I_S \models s\sigma \simeq t\sigma$, and by lemma 3.7 $s\sigma$ must be reducible by R_C (with a rule generated by an instance C' smaller than C).

Let this rule be the one that reduces $s\sigma$ in an innermost position p with $top(s\sigma|_p) = f$ as in case 2: no other rule reduces $s\sigma$ in a position p' below $\mathrm{maxpos}(s\sigma, p)$ with $top(u|_{p'}) \neq f$. Now a contradiction is obtained exactly as it is done in case 2, but always inferences by (AC-) superposition left are considered instead of superposition right. □

The following theorem states the refutational completeness of our inference system, provided the initial set of clauses has no constraints. It is well-known that with arbitrary initial clauses these methods are incomplete[2].

Theorem 3.12 *(Refutation Completeness of* BOAC*)*
Let S be the closure under BOAC *of a set S_0 of clauses with empty constraints. Then the empty clause is in S iff $S_0 \cup AC$ is inconsistent.*

[2] cf. [NR92b] for counter examples, and for a discussion about what kind of initial clauses can be allowed, like arbitrary constrained clauses without equality literals.

Proof *The left-to-right implication is trivial, since the inference rules of* BOAC *are sound, i.e.* S_0 *and* S *are logically equivalent modulo* AC*. For the right-to-left implication, suppose the empty clause is not in* S*. We prove that then* $S_0 \cup AC$ *is consistent, as it has an* AC*-model, namely* I_S*. From the previous lemma, we have that* $I_S \models irred(S)$ *and* $I_S \models AC$*. Since* S_0 *is contained in* S*, in particular* $I_S \models C\sigma$ *for each* C $[\![\ true\ ]\!]$ *in* S_0 *with* $x\sigma$ *irreducible wrt.* R_S *for each* x *in* $\mathcal{V}ars(C)$ *(which are existing instances of clauses in* S_0 *since these clauses have no constraints). But since* I_S *is a congruence containing* R_S*, it also satisfies the instances of* S_0 *that are reducible by* R_S*, since these instances follow from the irreducible ones and* R_S*. This means* $I_S \models S_0$*.* $\square$

Recall that $C\ [\![\ T\ ]\!]$ is the empty constrained clause if C is empty and T is satisfiable, and that our inference rules are sound also if no ordering constraints are added, but that the AC-equality constraints are essential for soundness. Therefore we only have to decide the satisfiability of the AC-equality part of T to know whether $C\ [\![\ T\ ]\!]$ implies the inconsistency of S_0 or not. It is an open problem whether the satisfiability of our kind of ordering constraints is decidable or not, but this is not needed here. We only need sufficiently powerful methods for detecting as many unsatisfiable ordering constraints as possible, which can be done applying the extension of our ordering to terms with variables.

Example 3.13 *If* f *is an* AC *symbol and e.g.* $a \succ c \succ d$ *then the set*

1.		$\rightarrow f(a,b) \simeq c$
2.		$\rightarrow f(f(a,a), f(b,b)) \simeq d$
3.	$f(c,c) \simeq d$	$\rightarrow$

is inconsistent[3]*. Now an* AC*-strict superposition right inference by* 1. *on* 2. *with conclusion* 4. $\rightarrow f(c,x) \simeq d\ [\![\ x =_{\mathrm{AC}} f(a,b)\ ]\!]$ *can be made (like normal superposition on the topmost* f *with a so-called* extension $f(f(a,b),x) \simeq f(c,x)$*). This seems insufficient to obtain a refutation: an inference with* 1. *seems to be needed* below x *in* 4*. But this can also be done by another* AC*-inference on* 4. *with* 1. *on the maximal* f*, producing* 5. $\rightarrow f(c,y) \simeq d\ [\![\ x =_{\mathrm{AC}} f(a,b) \wedge y =_{\mathrm{AC}} c\ ]\!]$*. This is why def. 3.3–3. considers in clause* 4. *the variable* x *instantiated with* $f(a,b)$ *to be variable irreducible (our results imply that* blocking*, i.e. forbidding inferences with reducible unifiers, is complete for* AC *iff "reducible" is replaced by "variable reducible"). Finally,* 5. *produces the empty clause with* 3.

Example 3.14 *The clause* $\rightarrow f(x,0) \simeq x$ *would usually be considered* self-extending*, i.e. no* AC*-superposition with this clause would be needed. But this is incompatible with basicness:* $f(0, f(0,0)) \simeq 0 \rightarrow$ *cannot be refuted by basic superposition on maximal positions ([Wer92]), since only* $x \simeq 0 \rightarrow\ [\![\ x =_{\mathrm{AC}} f(0,0)\ ]\!]$ *could be obtained.*

By our basic AC*-superposition rule, one gets* $f(y,x) \simeq 0 \rightarrow\ [\![\ x =_{\mathrm{AC}} 0 \wedge y =_{\mathrm{AC}} 0\ ]\!]$*, and then* $z \simeq 0 \rightarrow\ [\![\ x =_{\mathrm{AC}} 0 \wedge y =_{\mathrm{AC}} 0 \wedge z =_{\mathrm{AC}} 0\ ]\!]$*, and finally, by equality resolution, the empty clause* $\rightarrow\ [\![\ x =_{\mathrm{AC}} 0 \wedge y =_{\mathrm{AC}} 0 \wedge z =_{\mathrm{AC}} 0\ ]\!]$*.*

[3] This example is adapted from[BD89]. We thank Leo Bachmair for pointing this example to us and for his comments.

4 Compatibility with simplification and deletion

In the previous section, for simplicity reasons, we have only proved the refutational completeness of the inference system, i.e. that inconsistent sets closed under the inference rules will contain the empty clause. But in fact theorem 3.12 also holds for sets that are closed *up to redundant inferences*, the so-called saturated sets. Here we will define theorem proving derivations that systematically compute saturated sets of clauses, even if *redundant clauses* are deleted during the theorem proving process and redundant inferences are not considered. The empty clause will be deduced in such derivations starting from an inconsistent initial set of clauses without constraints. We will first consider abstract redundancy notions and give practical methods for simplification and deletion later on. In the following we consider only inferences by $BOAC$:

Definition 4.1 *(cf. [BG91, NR92b] for more details).*
Let S be a set of constrained clauses.

1. *A clause C $[\![\ T\ ]\!]$ is redundant in S if for every set of rules R s.t. $\Rightarrow_R\ \subseteq\ \succ$ and for every ground instance $C\sigma$ that is variable irreducible wrt. R, there exist ground instances $e_1 \dots e_m$ of AC and variable irreducible wrt. R ground instances $D_1 \dots D_n$ of clauses in S with $C\sigma \succ e_i$ and $C\sigma \succ D_i$ and $R \cup \{D_1, \dots, D_m, e_1, \dots, e_m\} \models C\sigma$.*

2. *An inference π with premise C $[\![\ T\ ]\!]$, or premises C' $[\![\ T'\ ]\!]$ and C $[\![\ T\ ]\!]$, and conclusion D $[\![\ T''\ ]\!]$ is redundant in S if for every set of rules R s.t. $\Rightarrow_R\ \subseteq\ \succ$ and for every ground inference of the form $\pi\sigma$ whose premises and conclusion are variable irreducible instances wrt. R, there exist ground instances $e_1 \dots e_m$ of AC and variable irreducible wrt. R ground instances $D_1 \dots D_n$ of clauses in S such that $C\sigma \succ e_i$ and $C\sigma \succ D_i$ and $R \cup \{D_1, \dots, D_m, e_1, \dots, e_m\} \models D\sigma$.*

3. *S is* saturated *if every inference with premises in S is redundant in S.*

4. *A* theorem proving derivation *is a sequence of sets of (constrained) clauses $S_0, S_1, \dots$ where each S_{i+1} is obtained from S_i by adding a clause following from $S_i \cup AC$ or by removing a clause that is redundant in S_i.*

5. *The set S_∞ of persistent clauses in $S_0, S_1, \dots$ is defined as $\cup_j(\cap_{k \succeq j} S_k)$.*

6. *A theorem proving derivation $S_0, S_1, \dots$ is fair if, for every inference π with persistent premises, there is some S_j s.t. π is redundant in S_j.*

Lemma 4.2 *If $S_0, S_1, \dots$ is a fair theorem proving derivation then S_∞ is saturated.*

Lemma 4.3 *Let S be a saturated set of constrained clauses not containing the empty clause. Then $I_S \models irred(S) \cup AC$.*

Proof *By adapting lemma 3.11 as done in [NR92b]: the conclusions of the inferences now need not belong to S, but instead the inferences are redundant, i.e. the conclusion is deducible from certain instances smaller than the maximal premise wich leads to a contradiction as in lemma 3.11.* □

Theorem 4.4 *Let $S_0, S_1, \ldots$ be a fair theorem proving derivation, where S_0 is a set of clauses with empty constraints. Then $S_0 \cup AC$ is inconsistent iff the empty clause is in some S_j. Moreover, if some S_j is saturated and does not contain the empty clause, then $I_{S_j} \models S_j \cup AC$.*

4.1 Practical simplification and deletion methods

Definition 4.5 *A variable x is* not lower bounded *by a constraint T if $T\sigma \equiv true$ implies $T\sigma' \equiv true$, for every pair of ground substitutions σ and σ', s.t. $x\sigma \succ x\sigma'$ and $y\sigma' = y\sigma$ for every other variable y in the domain of σ.*

Lemma 4.6 *Let S be a set of constrained clauses. A clause $C\,[\![\, T \,]\!]$ is redundant in S if for every ground instance $C\sigma$ of it, there exist ground instances $e_1 \ldots e_m$ of AC and ground instances $D_i\sigma_i$ of clauses $D_1\,[\![\, T_1 \,]\!] \ldots D_n\,[\![\, T_n \,]\!]$ in S, with $C\sigma \succ e_i$ and $C\sigma \succ D_i\sigma_i$, such that $\{e_1, \ldots, e_m, D_1\sigma_1, \ldots, D_n\sigma_n\} \models C\sigma$ and for every variable x in each D_i:*

1. *x is not lower bounded by T_i, or else*
2. *there is a variable $y \in \mathcal{V}ars(C)$ s.t. $x\sigma_i = y\sigma$, with $y \notin stvars(C, \sigma)$ and there is no AC-symbol f such that y only occurs in C immediately below f.*

A similar lemma as the previous one can be given for the redundancy of inferences. The lemma provides the conditions that have to be fulfilled by any *practical* method we want to use in this framework for proving that during the theorem proving process a certain clause can be deleted (e.g. because it is a tautology, or because it is subsumed by another one) or replaced by a simpler one (by demodulation or other simplification techniques). Indeed, we can construct counter examples for simplification methods that do not fulfil these conditions (cf. [NR92b] for the non-AC case).

Because of lack of space we do not discuss here which adapted versions for subsumption, simplification, etc. fall into the conditions of the previous lemma (this is a simple exercise; we are not aware of any such sound practical method that cannot be reformulated to follow from the previous lemma). Let us only mention that it is also possible to propagate (called to *weaken* in [NR92b]) parts of the constraints of the clauses used in redundancy proofs (which may require computing AC-unifers) to make conditions 1. and 2. of the previous lemma to hold.

5 Saturated sets and Knuth-Bendix completion

Suppose S is a finite saturated set without the empty clause, obtained from an initial set without constraints (in the following, such a set will be simply called saturated). Since $I_S \models S$, obtaining such an S proves the consistency of the theory (one can normally only prove inconsistencies), and on the other hand it is an efficient tool for theorem proving in this theory, since no inferences have to be computed between clauses of S. In fact, in some cases, depending on the syntactic properties of S, decision procedures for the theory are obtained. This is the case e.g. for saturated sets of equations E, which are convergent for both *rewriting modulo AC*, denoted $\rightarrow_{E/\mathrm{AC}}$, and for *extended AC-rewriting* denoted $\rightarrow_{E\backslash\mathrm{AC}}$. Let $l\sigma \simeq r\sigma$ be an instance of an equation of E with $l\sigma \succ r\sigma$. Then $s \rightarrow_{E/\mathrm{AC}} u[r\sigma]$ if $s =_{\mathrm{AC}} u[l\sigma]$. Furthermore, extended AC-rewriting is defined as $s[u] \rightarrow_{E\backslash\mathrm{AC}} s[r\sigma]$ if $u =_{\mathrm{AC}} l\sigma$ and $s[u] \rightarrow_{E\backslash\mathrm{AC}} s[f(r,x)\sigma]$ if $u =_{\mathrm{AC}} f(l,x)\sigma$.

Theorem 5.1 *Let E be a saturated set of constrained equations. For all terms s and t*

$$E \cup AC \models s \simeq t \quad \textit{iff} \quad s \rightarrow^*_{E/\mathrm{AC}} \circ \leftrightarrow^*_{\mathrm{AC}} \circ \leftarrow^*_{E/\mathrm{AC}} t \quad \textit{iff} \quad s \rightarrow^*_{E\backslash\mathrm{AC}} \circ \leftrightarrow^*_{\mathrm{AC}} \circ \leftarrow^*_{E\backslash\mathrm{AC}} t.$$

Proof *We prove that every ground term s (possibly with new Skolem constants to which our ordering $\succ$ can be extended) can be rewritten into some minimal (wrt. $\succ$) representative of its $(E \cup AC)$-congruence class (note that all minimal representatives of the same class are AC-equal since $\succ$ is total). We proceed by induction wrt. $\succ$, i.e. it suffices to prove the reducibility wrt. $\rightarrow_{E/\mathrm{AC}}$ and $\rightarrow_{E\backslash\mathrm{AC}}$ of non-minimal s.*

Let t be such a minimal representative of the congruence class of s. If $s \succ t$, the only inference rules that can be applied in a refutation of $s \simeq t \rightarrow$ are (AC and non-AC) strict superposition left steps on s with some equation $l \simeq r$ $[\![T]\!]$ of E. Then the conclusion is a clause C of the form $s[r]_p \simeq t \rightarrow$ $[\![T \wedge s \succ t \wedge l \succ r \wedge s|_p =_{\mathrm{AC}} l]\!]$ (in the case of a non-AC step) or $s[f(r,x)]_p \simeq t \rightarrow$ $[\![T \wedge s \succ t \wedge l \succ r \wedge s|_p =_{\mathrm{AC}} f(l,x)]\!]$ (in an AC step). Such a step must exist, since $E \cup AC \models s \simeq t$, and a ground instance $C\sigma$ must exist such that the constraint is true, which implies that $l\sigma \succ r\sigma$. But then s is reducible by extended AC-rewriting, as $s|_p =_{\mathrm{AC}} l\sigma$ or $s|_p =_{\mathrm{AC}} f(l,x)\sigma$ for some p and, for rewriting modulo AC, we have $s =_{\mathrm{AC}} u[l\sigma]_p$. □

Let us remark that if r has no "extra variables" (variables that are not in l), then AC-matching provides such a ground σ and we can check whether σ fulfils the equality constraint part T' of T (the ordering part can be ignored). Otherwise, it suffices to instantiate the extra variables with the adequate mgu of T', and the remaining variables with the smallest constant (this provides a smaller *ground* term of the same $(E \cup AC)$-congruence class iff such a term exists). The mgu's can also be computed once and for all for E before rewriting (for this particular purpose, computing AC-unifiers may even be unnecessary, but this has to be studied in detail).

A similar decision result (by refutation or by conditional rewriting) holds for ground queries $S \cup AC \models s_1 \simeq t_1 \wedge \ldots \wedge s_n \simeq t_n$ if S contains only Horn clauses $\Gamma \rightarrow$ $[\![T]\!]$ (which need not to be used) and Horn clauses $\Gamma \rightarrow s \simeq t$ $[\![T]\!]$ where for each mgu θ of the equality part of T and for all ground instances with $\theta\sigma$ the positive equation $s\theta\sigma \simeq t\theta\sigma$ is strictly maximal and if $s\theta\sigma \succ t\theta\sigma$ then $\mathcal{V}ars(s\theta) \supseteq \mathcal{V}ars(\Gamma\theta)$.

References

[BD89] Leo Bachmair and Nachum Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 2 and 3(67):173–201, 1989.

[BG91] Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. Technical Report MPI-I-91-208, Max-Planck-Institut für Informatik, Saarbrücken, August 1991. To appear in Journal of Logic and Computation.

[BGLS92] Leo Bachmair, Harald Ganzinger, Christopher Lynch, and Wayne Snyder. Basic paramodulation and superposition. In Deepak Kapur, editor, *11th International Conference on Automated Deduction*, LNAI 607, pages 462–476, Saratoga Springs, New York, USA, June 15–18, 1992. Springer-Verlag. To appear in Information and Computation.

[BS93] Franz Baader and Jörg Siekmann. Unification theory. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press (to appear), 1993.

[Dom92] Eric Domenjoud. A technical note on AC-unification. the number of minimal unifiers of the equation $\alpha x_1 + \ldots + \alpha x_p = \beta y_1 + \ldots + \beta y_q$. *Journal of Automated Reasoning*, 8(1):39–44, 1992.

[JK86] Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15:1155–1194, 1986.

[KKR90] Claude Kirchner, Hélène Kirchner, and Michaël Rusinowitch. Deduction with symbolic constraints. *Revue Française d'Intelligence Artificielle*, 4(3):9–52, 1990.

[KN92] Deepak Kapur and Paliath Narendran. Complexity of unification problems with associative commutative operators. *Journal of Automated Reasoning*, 9:261–288, 1992.

[NR91] Paliath Narendran and Michael Rusinowitch. Any ground associative commutative theory has a finite canonical system. In *Fourth int. conf. on Rewriting Techniques and Applications*, LNCS 488, pages 423–434, Como, Italy, April 1991. Springer-Verlag.

[NR92a] Robert Nieuwenhuis and Albert Rubio. Basic superposition is complete. In B. Krieg-Brückner, editor, *European Symposium on Programming*, LNCS 582, pages 371–390, Rennes, France, February 26–28, 1992. Springer-Verlag.

[NR92b] Robert Nieuwenhuis and Albert Rubio. Theorem proving with ordering constrained clauses. In Deepak Kapur, editor, *11th CADE*, LNAI 607, pages 477–491, Saratoga Springs, New York, 1992. Extended version to appear in Journal of Symbolic Computation.

[PS81] G.E. Peterson and M.E. Stickel. Complete sets of reductions for some equational theories. *Journal Assoc. Comput. Mach.*, 28(2):233–264, 1981.

[RN93] Albert Rubio and Robert Nieuwenhuis. A precedence-based total AC-compatible ordering. In C. Kirchner, editor, *5th International Conference on Rewriting Techniques and Applications*, LNCS 690, pages 374–388, Montreal, Canada, June 16–18, 1993. To appear in Theoretical Computer Science.

[Rub94] Albert Rubio. Automated deduction with ordering and equality constrained clauses. PhD. Thesis, Technical University of Catalonia, Barcelona, Spain, 1994.

[RV93] Michael Rusinowitch and Laurent Vigneron. Automated deduction with associative commutative operators. Rapport de Recherche 1896, Institut National de Recherche en Informatique et en Automatique, INRIA-Lorraine, May 1993. To appear in J. of Applicable Algebra in Engineering, Communication and Computation.

[Vig94] Laurent Vigneron. Associative Commutative Deduction with constraints. In Allan Bundy, editor, *12th International Conference on Automated Deduction*, LNAI, Nancy, France, June 1994. Springer-Verlag.

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