J. Symbolic Computation (1996) **11**, 1–21

Paramodulation with Built-in AC-Theories and Symbolic Constraints

ROBERT NIEUWENHUIS AND ALBERT RUBIO †

Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain

E-mail: {roberto,rubio}@lsi.upc.es.

(Received 16 June 1995)

Complete paramodulation strategies are developed for clauses with symbolic constraints and built-in associativity and commutativity (AC) properties for a subset of the function symbols. Apart from the reduced search space due to the inherited ordering and unification restrictions of the inferences (cf. Nieuwenhuis and Rubio, 1995), symbolic constraints turn out to be especially useful in the context of built-in equational theories E. In every inference, instead of producing as many conclusions as minimal E-unifiers of two terms s and t, one single conclusion is generated with an additional E-unification problem $s = t$ kept in its constraint. In the AC-case developed here (the most extensively studied built-in theory), the computation of the —doubly exponentially many— AC-unifiers can thus be completely avoided. These results are also applied here to more efficient strategies (sometimes decision procedures) in theories expressed by finite *saturated* sets of clauses and, in particular, to rewriting and Knuth-Bendix completion modulo AC.

1. Introduction

Resolution methods can of course be applied in the presence of the equality predicate, provided the equivalence and replacement axioms of equality are added to the set of clauses. However, it is well-known that many times a better performance is achieved by the special equality-oriented inference rule of paramodulation (Robinson and Wos, 1969), and its refinements, like rewriting and Knuth-Bendix completion-based techniques in pure equational reasoning, and ordered (Hsiang and Rusinowitch, 1991; Bachmair and Ganzinger, 1994) and basic paramodulation and superposition (Nieuwenhuis and Rubio, 1995; Bachmair *et al.*, 1995) for general clauses.

These techniques for "building in" the equality predicate apply to any set of clauses, but again special treatments for some equational subset of the axioms are usually worthwhile. Historically, these special treatments were motivated by the fact that equations like the commutativity axiom $f(x, y) \simeq f(y, x)$ cannot be oriented into a (terminating) rewrite rule. Although this problem was overcome by the ordered rewriting approach, there are several other good reasons for having specific methods for particular equations.

† A preliminary short version of this work appeared in proc. CADE'94. Both authors are partially supported by the Esprit Working Group CCL, ref. 6028

0747–7171/90/000000 + 00 $03.00/0 © 1996 Academic Press Limited

On one hand, some axioms generate many slightly different permuted versions of clauses, and for efficiency reasons it is often better to treat all these clauses together as a single one representing the whole class. On the other hand, special treatments can avoid non-termination of completion procedures, like it happens with $f(a,b) \simeq c$ in the presence of associativity and commutativity axioms for f. Also, as observed in (Bachmair, 1991), some equations like the commutativity axiom are more naturally viewed as "structural" axioms (defining a congruence relation on terms) rather than as "simplifiers" (defining a reduction relation). In this way deduction methods can be extended to deal with congruence classes of terms instead of single terms, i.e. working with a *built-in* equational theory E, and performing rewriting and completion with special E-matching and E-unification algorithms.

The first results on rewriting *modulo* E were given by Lankford and Ballantine (1977) and extended *E-rewriting* was defined by Peterson and Stickel (1981). Many E-completion and refutation procedures were developed e.g. in (Lankford and Ballantyne, 1977; Huet, 1980; Peterson and Stickel, 1981; Jouannaud, 1983; Jouannaud and Kirchner, 1986; Bachmair and Dershowitz, 1987; Bachmair, 1991; Paul, 1992; Rusinowitch and Vigneron, 1993), some of them for specific sets E and others for wider classes of sets.

Most of the attention has always been devoted to the case where E consists of axioms of associativity (A) and commutativity (C), which occur very frequently in practical equational (or clausal) specifications, and are well-suited for being built in due to their permutative nature. In such methods, (equational) paramodulation modulo AC is defined:

$$\frac{s' \simeq t' \qquad s \simeq t}{(s[t']_p \simeq t)\sigma} \quad \forall\, \sigma \in cU_{AC}(s|_p, s')$$

i.e. one conclusion is added for each σ in $cU_{AC}(s|_p, s')$, a minimal complete set of AC-unifiers of $s|_p$ and s'. This has motivated a huge amount of research on computing complete sets of AC-unifiers (e.g. Stickel, 1981; Fages, 1987; Fortenbacher, 1987; Bürckert *et al.*, 1988; Kirchner, 1989; Lincoln and Christian, 1989; Boudet *et al.*, 1990; Domenjoud, 1992a; Kapur and Narendran, 1992b; see Baader and Siekmann, 1993 for a recent survey on (AC-)unification). One drawback is the complexity of AC-unification: there may be doubly exponentially many AC-unifiers for two terms (Domenjoud, 1992b), and therefore as many conclusions in an inference; e.g. a minimal complete set for $x + x + x$ and $y_1 + y_2 + y_3 + y_4$ contains more than a million unifiers.

In this paper we overcome these drawbacks by working with clauses with symbolic constraints (Kirchner *et al.*, 1990; Nieuwenhuis and Rubio, 1992; Rubio, 1994; Nieuwenhuis and Rubio, 1995). A constrained clause $C \;[\![\, T \,]\!]$ is a shorthand for the set of ground instances of the clause part C satisfying the constraint T. In a constrained equation $f(x) \simeq a \;[\![\, x = g(y) \,]\!]$, 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 (Kirchner *et al.*, 1990) required 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 $f(g(y)) \simeq a \;[\![\, true \,]\!]$. Avoiding propagation is essential for exploiting the constraints. By doing so for (syntactic) equality constrained clauses, in (Nieuwenhuis and Rubio, 1992) the completeness of *basic superposition* was proved (independently also in Bachmair *et al.*, 1992), and by adding *ordering constraints* the search space was further reduced by *inheriting* the ordering restrictions of the inference rules (Nieuwenhuis and Rubio, 1995). E.g. constrained

(equational) superposition can be expressed like:

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

This provides an elegant and powerful representation for ordered inference rules, where 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 (see Nieuwenhuis and Rubio, 1995 and Rubio, 1994 for complete developments of this framework). 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.

Such symbolic constraints turn out to be especially interesting when working modulo equational theories. Here we prove the refutation completeness of a superposition-based inference system, where each inference has *one single conclusion* with an additional equality $s = t$ in its constraint, instead of one conclusion for each minimal AC-unifier of s and t. This also eliminates the need of computing AC-unifiers in AC-deduction (or completion) methods. A clause C with an AC-equality constraint T of the form $s_1 = t_1 \wedge \ldots \wedge s_n = t_n$ can be proved redundant by means of efficient incomplete methods detecting cases of unsatisfiability of T. Only if C is the empty clause one has to actually *decide* the AC-unifiability of T (which is NP-complete, cf. Kapur and Narendran, 1992a) in order to know whether an inconsistency has been derived or not.

In our completeness proofs we apply an essential ingredient which we gave for this purpose in (Rubio and Nieuwenhuis, 1995): an AC-compatible simplification ordering that is total (up to AC-equality) on ground terms. This AC-RPO 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 (Narendran and Rusinowitch, 1991) is defined only on ground terms). This suffices for detecting unsatisfiable ordering contraints and, hence, redundant clauses. Since the ordering restrictions are not related to the soundness of our inference rules, an inconsistency is derived when the empty clause is generated even if its ordering constraint (but not its equality constraint) is unsatisfiable. Therefore, the decision procedure of (Comon *et al.* , 1995) for the satisfiability of AC-RPO constraints will be applied in this context only when one wants to put the maximal effort in detecting as many redundant clauses as possible.

The proof techniques applied here 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 (1994; see also Pais and Peterson, 1991 for a similar model construction proof technique). These techniques are adapted here to AC-deduction, similarly to Wertz's work (Wertz, 1992), although he deals with explicit so-called *extended* clauses, while we simulate them by means of specific AC-inference rules, like in (Rusinowitch and Vigneron, 1993). In our opinion, the new proof technique given here for building in AC is interesting in itself (apart from the advantages due to the symbolic constraints) because of its simplicity.

The first results on (almost basic) constrained deduction modulo AC were reported by Laurent Vigneron. In a recent version of his work (Vigneron, 1994), the computation

of AC-unifiers is also avoided (by applying our notion of irreducibility, defin. 4.3), and some further refinements as well as examples by an implementation of these techniques are given. His proof methods are completely different from ours and based on transfinite semantic trees as in (Rusinowitch and Vigneron, 1993). In a comparison of his work with ours, it turns out that the same refinements for reducing the number of inferences can be obtained in both frameworks, but that our version of the model generation method provides simpler proofs than the semantic tree method. Furthermore, the known extensions to the model generation method for constrained paramodulation like *redex orderings* and *variable abstraction* (Bachmair *et al.*, 1995) can also be smoothly incorporated here. Regarding simplification and other redundancy methods, the *abstract* redundancy notions given here express sharp bounds on the existing *concrete* redundancy methods (like the ones given in Vigneron, 1994, which indeed fit into our abstract ones).

2. Basic notions and terminology

An *equation* is a multiset of terms $\{s, t\}$, which will be written $s \simeq t$. A first-order clause is a pair of finite multisets of equations Γ (the *antecedent*) and Δ (the *succedent*), denoted by $\Gamma \rightarrow \Delta$. By *(ordering and equality) constraints* we mean quantifier-free first-order formulae built over the binary predicate symbols $\succ$ and $=$ relating terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $=$ is interpreted as AC-equality, and $\succ$ denotes an AC-compatible simplification ordering on ground terms that is total on the AC-congruence classes. We say that a ground substitution σ *satisfies* a constraint T if $T\sigma$ is (or evaluates to) true in this sense.

We extend $\succ$ to an ordering on ground equations (in fact, to their *occurrences* in clauses) and to clauses, s.t. terms in the antecendent get a slightly higher complexity than in the succedent. An occurrence of an equation $t \simeq t'$ in an antecendent is (the two-fold multiset) $\{\{t, t'\}\}$, and in a succedent it is $\{\{t\}, \{t'\}\}$. Now the two-fold multiset extension of $\succ$ is total on (AC-distinct occurrences of) ground equations, and the three-fold multiset extension of $\succ$ on $\Gamma \cup \Delta$ is a total ordering on (AC-distinct) ground clauses $\Gamma \rightarrow \Delta$. We will ambiguously use $\succ$ to denote all these orderings on ground terms, equations and clauses. An equation e is called *maximal* in a ground clause C if there is no equation e' in C such that $e' \succ e$ and *strictly maximal* if there is no e' with $e' \succeq e$ (i.e. $e' \succ e$ or $e' =_{\mathrm{AC}} e$).

An (AC-) interpretation I is a congruence on ground terms (satisfying the AC-axioms for all AC-operators). It satisfies a ground clause $\Gamma \rightarrow \Delta$, denoted $I \models \Gamma \rightarrow \Delta$, if $I \not\supseteq \Gamma$ or $I \cap \Delta \neq \emptyset$. An interpretation I satisfies (is a model of) a constrained clause $C \;[\![\, T \,]\!]$, denoted $I \models C \;[\![\, T \,]\!]$, if it satisfies every ground instance of $C \;[\![\, T \,]\!]$, i.e. every $C\sigma$ such that σ is ground and $T\sigma$ is true. Therefore, clauses with unsatisfiable constraints are tautologies, and $C \;[\![\, T \,]\!]$ is the *empty clause* only if C is empty and T is satisfiable. I satisfies a set of clauses S, denoted by $I \models S$, if it satisfies every clause in S. A clause C follows from a set of clauses S (denoted by $S \models C$), if C is satisfied by every model of S. For dealing with non-equality predicates, atoms A can be expressed by equations $A \simeq \mathit{true}$ where *true* is a special symbol (minimal in $\succ$).

If C is a ground clause and S is a set of clauses, then we denote by $S^{\prec C}$ (resp. $S^{\preceq C}$) the set of ground instances of clauses in S that are smaller wrt. $\succ$ than C (resp. smaller than or equal to C).

We use the definitions of (Dershowitz and Jouannaud, 1990) for rewriting-related notions. However, to avoid confusion with the arrow $\rightarrow$ of clauses, we denote ground rewrite

rules (ground equations $s \simeq t$ with $s \succ t$) by $s \Rightarrow t$. The congruence generated by a set of equations (or rewrite rules) E (which is an interpretation) will be denoted by E^*.

It is well-known that a term s can be *flattened* by removing all AC-operators f that are immediately below another f. For example, if f and g are AC-operators, then $h(f(f(a,a),f(b,g(c,g(d,e)))))$ is flattened into $h(f(a,a,b,g(c,d,e)))$. We do not use flattening in this paper, except for illustrating the following. The symbols that are not removed under flattening are in a *maximal position*: if p is a position in a ground term s, we define $maxpos(s,p)$ to be p if $top(s|_p)$ is not an AC-operator and else $maxpos(s,p)$ is the maximal prefix p' of p such that $p' = \lambda$ or $p' = p'' \cdot n$ with $top(s|_{p''}) \neq top(s|_p)$. Let s and t be two AC-equal terms, i.e. $s =_{\mathrm{AC}} t$. Then their flattened forms are equal up to permutation of arguments of AC-operators: a one-to-one correspondance can be established in this way between maximal positions in s and in t. We will sometimes speak about the *corresponding* position in t of some maximal position in s. Note that if u and v are subterms at corresponding maximal positions of resp. s and t, then $u =_{\mathrm{AC}} v$. Moreover, if $f(t_1, \ldots t_n)$ is the term resulting of applying flattening only at top-most position then *top-flattening* of a term t, denoted $tf(t)$ is $\{t_1, \ldots, t_n\}$. For example, if f is an AC-operator, $tf(f(a,f(g(f(f(a,b),c)),f(a,b))))$ is $\{a, g(f(f(a,b),c)), a, b\}$.

3. Inference rules

DEFINITION 3.1. *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 the superposition rules include $s \succ t$ and $s' \succ t'$ and they also encode (cf. Nieuwenhuis and Rubio, 1995) maximality wrt. $\succ$ of the equations $s \simeq t$ and $s' \simeq t'$ in the premise to which they belong, which becomes strict maximality if they belong to the succedent. Similarly, in equality resolution OC encodes maximality of $s \succ t$, and in factoring $s \succ t$, $s' \succ t'$ and strict maximality of $s \succ t$.*

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. The new variable x introduced in an inference by AC-strict superposition (right or left) is a so-called extension variable *for f, i.e. it belongs to a set of fresh variables $\mathcal{X}_f$ that exists for each AC-function symbol f. 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 the next section.

1 strict superposition right:

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

2 strict superposition left:

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

3 equality resolution:

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

4 factoring:

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

5 AC-strict superposition right:

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

6 AC-strict superposition left:

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

7 AC-top-superposition:

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

4. Refutation completeness

DEFINITION 4.1. *Let $C \;[\![\, T \,]\!]$ 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 $C \;[\![\, T \,]\!]$.*

If moreover $C\sigma$ is of the form $\Gamma \rightarrow \Delta, s \simeq t$ where $top(s)$ is an AC symbol f, and $s \simeq t$ is the strictly maximal equation in $C\sigma$ with $s \succ t$, then for every ground term v, the clause $\Gamma \rightarrow \Delta, f(s, v) \simeq f(t, v)$ is called an extended (ground) instance of $C \;[\![\, T \,]\!]$ wrt. $C\sigma$ (with the context v).

Sometimes, to distinguish between extended instances and the usual instances, we will call the latter non-extended instances.

Similarly, if π is an inference (by BOAC*) with premises $C_1 \;[\![\, T_1 \,]\!], \ldots C_n \;[\![\, T_n \,]\!]$ and conclusion $C \;[\![\, T \,]\!]$ and σ is a ground substitution satisfying T, then the inference $\pi\sigma$ with premises $C_1\sigma, \ldots, C_n\sigma$ and conclusion $D\sigma$ is a* ground instance with *σ of π.*

DEFINITION 4.2. *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 4.3. *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 $x \in \mathcal{X}_f$ *for 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 wrt.* R.

Here point 2. is based on the irreducibility notion of (Bachmair *et al.*, 1995), 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 4.20. In preliminary versions of our work, in point 3 x was required to only occur in C immediately below some AC-symbol f, which is less restrictive than to be in some $\mathcal{X}_f$. This improvement to our notion of irreducibility was introduced in (Vigneron, 1994). This is interesting since inferences with constrained clauses with only reducible instances are not needed (see lemma 4.25), and therefore having a stronger notion of irreducibility allows one to avoid more inferences.

DEFINITION 4.4. *Let* R *be a set of ground rewrite rules,* $C \,[\![\, T \,]\!]$ *a constrained clause,* S *a set of constrained clauses and* π *an inference.*

1 The set of variable irreducible (wrt. R*) ground instances of* $C \,[\![\, T \,]\!]$ *is defined by* $Ir_R(C \,[\![\, T \,]\!]) = \{C\sigma \mid T\sigma \equiv true \wedge \sigma \text{ ground} \wedge C\sigma \text{ variable irreducible wrt. } R\}$.
2 Similarly, $Ir_R(S) = \{C\sigma \mid C \,[\![\, T \,]\!] \in S \wedge C\sigma \in Ir_R(C \,[\![\, T \,]\!])\}$.
3 $Ir_R(\pi)$ *is the set of ground instances* $\pi\sigma$ *of* π *such that* $C\sigma \in Ir_R(C \,[\![\, T \,]\!])$ *for each* $C \,[\![\, T \,]\!]$ *that is premise or conclusion of* π.

Now we define the set of rewrite rules R_S for a set S by induction. Each instance C may *generate* a rule depending on the set R_C of rules generated by smaller instances:

DEFINITION 4.5. *Let* S *be a set of constrained clauses. Now for each ground extended and non-extended instance* C *of a constrained clause in* S*, we inductively define the cases in which* C *generates certain ground rewrite rules, in terms of the set* R_C *of rules generated by instances smaller (wrt.* $\succ$*) than* C.

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* $(R_C \cup AC_C)^*$.

Let C *be a ground clause of the form* $\Gamma \rightarrow \Delta, s \simeq t$ *where* $s \simeq t$ *is strictly maximal (wrt.* $\succ$*) in* C *and* $s \succ t$.

1 If C *is a (non-extended) instance of a clause in* S *that is variable irreducible wrt.* R_C *then it generates the set of rules* $\{u \Rightarrow t \mid u =_{\mathrm{AC}} s\}$ *if:*

(a) $I_C \not\models C$,
(b) $I_C \not\models t \simeq t'$*, for every* $s' \simeq t'$ *in* Δ *with* $s' =_{\mathrm{AC}} s$*, and*
(c) v *is irreducible by* R_C*, for all ground terms* v *with* $v =_{\mathrm{AC}} s$

2 If C *is an extended instance of a clause in* S *wrt. 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* $I_S = (R_S \cup AC_S)^*$.

† To ensure that in this inductive definition all irreducible instances of AC are generated, even if there is some greatest instance C of S (e.g. when all clauses in S are ground), only for this definition it is supposed that the clause $\rightarrow x \simeq x \,[\![\, true \,]\!]$ is in S (note that this clause cannot generate any rule).

Note that by construction (and by AC-compatibility of $\succ$), for all rules $l \Rightarrow r$ that are generated, we have $l \succ r$.

LEMMA 4.6. *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 $I_C \supseteq I_D$.*

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, by AC-compatibility and transitivity of $\succ$, $s' \succ r$, i.e. $s \simeq s' \succ l \simeq r$. Therefore $s \simeq s' \succ D'$, which contradicts $s \simeq s' \in AC_{D'}$. From the two previous points $I_C \supseteq I_D$ follows. □

LEMMA 4.7. *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 $l =_{\text{AC}} l'$ then r and r' are the same term.
4 There are no overlaps between left hand sides of rules of R_S.
5 If $u =_{\text{AC}} v$ for ground terms u and v, then u is reducible by R_S iff v is.

PROOF.

1 By definition $C \succ D$ implies $l \succ l'$ or $l =_{\text{AC}} l'$, but $l =_{\text{AC}} l'$ cannot happen since the rule $l \Rightarrow r$ has been generated.
2 Let D be $\rightarrow s_1 \simeq s_2$. Then D is irreducible wrt. R_D and moreover, for all rules $l \Rightarrow r$ in $R_S \setminus R_D$, since $s_1 =_{\text{AC}} s_2$, $l \succ r$ and $l \simeq r \succeq s_1 \simeq s_2$ (as $l \simeq r$ is maximal in its clause), we have $l \succ s_1$ and $l \succ s_2$. Therefore, as $\succ$ fulfills the subterm property, $l \neq_{\text{AC}} s_1|_p$ and $l \neq_{\text{AC}} s_2|_p$ for all positions p.
3 Since $l \Rightarrow r$ and $l' \Rightarrow r'$ are in R_S, by construction, there are no other rules $l \Rightarrow r''$ or $l' \Rightarrow r''$ for any r''. If $l \Rightarrow r$ (or $l' \Rightarrow r'$) is a non-extended rule then, by definition, apart from $l \Rightarrow r$, all rules $u \Rightarrow r$ with $u =_{\text{AC}} l$ have been generated and hence r and r' must be the same term. Otherwise, if $l \Rightarrow r$ is an extended rule then $r \neq r'$ leads to a contradiction. By definition, as $l' \Rightarrow r$ was not generated, the rule $l' \Rightarrow r'$ was generated before the rule $l \Rightarrow r$, but as $l \Rightarrow r'$ was not generated either, it must be because the rule $l \Rightarrow r$ was generated before the rule $l' \Rightarrow r'$, which leads to a contradiction.
4 Let $l \Rightarrow r$ be a rule in R_S generated by a (possibly extended) instance D of a clause in S. Then l is irreducible wrt. R_D and, by property 1, $l' \succ l$ for all $l' \Rightarrow r' \in R_S \setminus (R_D \cup \{l \Rightarrow r\})$, and hence (by the subterm property of $\succ$) $l|_p \neq_{\text{AC}} l'$ for all positions p. Therefore there are no overlappings between rules in R_S.
5 Let v be reducible by a rule $l \Rightarrow t$ with $l =_{\text{AC}} s$ or $l \Rightarrow f(t, w)$ with $l =_{\text{AC}} f(s, w)$, generated by a non-extended instance $\Gamma \rightarrow \Delta, s \simeq t$ or by an extended (with w) instance $\Gamma \rightarrow \Delta, f(s, w) \simeq f(t, w)$, such that the rule $s \Rightarrow t$ has been generated. Therefore $v|_p = l$ for some position p.
Let $maxpos(v, p)$ be p_1. Then either $v|_{p_1} =_{\text{AC}} s$ or $v|_{p_1} =_{\text{AC}} f(s, w')$ for some w' (note that if $p \neq p_1$ then $v|_{p_1} =_{\text{AC}} f(v|_p, w'')$ for some w'', where $w' = w''$ if the non-extended rule is considered or $w' = f(w, w'')$ otherwise). Therefore for

the corresponding maximal position p_2 in u, i.e. $v|_{p_1} =_{\mathrm{AC}} u|_{p_2} = l'$, since $l' =_{\mathrm{AC}} s$ or $l' =_{\mathrm{AC}} f(s, w')$ and the rule $s \Rightarrow t$ was generated, either the rule $l' \Rightarrow t$ or $l' \Rightarrow f(t, w')$ has been generated or l' is reducible by R_S, which implies in any case that u is reducible by R_S.

□

LEMMA 4.8. *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 4.9. *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 $I_S \supseteq \Gamma\sigma$ and $I_S \cap \Delta\sigma = \emptyset$.*

PROOF. Since C generates rules, $I_C \supseteq \Gamma\sigma$ and then also $I_S \supseteq \Gamma\sigma$. It remains to be shown that $I_S \cap \Delta\sigma = \emptyset$.

We know $I_C \cap \Delta\sigma = \emptyset$ (since C generates rules) 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, by definition 4.5, C cannot generate any rule. □

LEMMA 4.10. *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. □

DEFINITION 4.11. *Let R be a set of ground rewrite rules.*

1 *A constrained clause $D \llbracket T \rrbracket$ is* redundant *wrt. R in a set of constrained clauses S if $Ir_R(S)^{\preceq C} \cup AC^{\preceq C} \cup R \models C$ for all C in $Ir_R(D \llbracket T \rrbracket)$.*

2 *An inference* π *is* redundant *wrt.* R *in a set* S *if* $Ir_R(S)^{\prec C} \cup AC^{\prec C} \cup R \models D$ *for all inferences in* $Ir_R(\pi)$ *with maximal premise* C *and conclusion* D.

DEFINITION 4.12. *A set of constrained clauses* S *is* saturated *if all inferences by* BOAC *with premises in* S *are redundant wrt.* R_S *in* S.

LEMMA 4.13. *Let* S *be a saturated set of constrained clauses not containing the empty clause. Then* $I_S \models Ir_{R_S}(S) \cup AC$.

PROOF. We will derive a contradiction from the existence of a minimal (wrt. $\succ$) instance C in $Ir_{R_S}(S) \cup AC$ 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 (by lemma 4.7 point 5). 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, $f(u_2, u_1)$ must be also reducible at top-most position by R_S. Hence (by lemma 4.7 point 3) there is a rule $f(u_2, u_1) \Rightarrow r$ in R_S, 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 $C \notin AC_S$, as before one of $f(f(u_1, u_2), u_3)$ and $f(u_1, f(u_2, u_3))$ must be 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 $w \Rightarrow t\sigma$ with $w =_{\mathrm{AC}} s\sigma$ or $w \Rightarrow f(t\sigma, v)$ with $w =_{\mathrm{AC}} f(s\sigma, v)$, 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 $s_1 =_{\mathrm{AC}^{\prec C}} f(t\sigma, v_1)$ (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 $w' \Rightarrow t'\sigma$ with $w' =_{\mathrm{AC}} s'\sigma$ or $w' \Rightarrow f(t'\sigma, v')$ with $w' =_{\mathrm{AC}} f(s'\sigma, v')$, generated by an instance with σ (possibly extended with the context v') of a clause D' of the form $\Gamma' \rightarrow \Delta', s' \simeq t'$ (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 $s_2 =_{\mathrm{AC}^{\prec C}} f(t'\sigma, v_2)$.

A. If $tf(v_1) \not\subseteq tf(v_2)$ and $tf(v_2) \not\subseteq tf(v_1)$ then let v_1', v_2' and v_3 be ground terms s.t. $f(t\sigma, v_1) =_{\mathrm{AC}^{\prec C}} f(f(t\sigma, v_1'), v_3)$, $f(t'\sigma, v_2) =_{\mathrm{AC}^{\prec C}} f(f(t'\sigma, v_2'), v_3)$ and $tf(v_1') \cap tf(v_2') = \emptyset$ (if $tf(v_1) \cap tf(v_2) = \emptyset$ then $v_1 = v_1'$, $v_2 = v_2'$ and we do not need v_3). Therefore $f(s\sigma, v_1') =_{\mathrm{AC}} f(s'\sigma, v_2')$.

If $s'\sigma =_{\mathrm{AC}} v'_2$ (and hence $s\sigma =_{\mathrm{AC}} v'_1$) then we have, by minimality of C, $I_S \models f(t\sigma, t'\sigma) \simeq f(t'\sigma, t\sigma)$ and, since $s'\sigma =_{\mathrm{AC}\prec C} v'_2$ and $s\sigma =_{\mathrm{AC}\prec C} v'_1$, $I_S \models f(f(t\sigma, v'_1), v_3) \simeq f(f(t'\sigma, v'_2), v_3)$, but then $f(f(u_1, u_2), u_3) \rightarrow_{R_S} s_1 =_{\mathrm{AC}\prec C} f(f(t\sigma, v'_1), v_3)$ and $f(u_1, f(u_2, u_3)) \rightarrow_{R_S} s_2 =_{\mathrm{AC}\prec C} f(f(t'\sigma, v'_2), v_3)$ contradict $I_S \not\models C$. Otherwise, if $s'\sigma \neq_{\mathrm{AC}} v'_2$ there is an inference by AC-top-superposition between D' and D:

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

whose conclusion D_1 has an instance with a ground substitution θ defined by $\theta = \sigma \cup \{x \mapsto v'_1, x' \mapsto v'_2\}$ (we are supposing wlog. $D\theta \succ D'\theta$).

$D_1\theta$ is variable irreducible wrt. R_S since $D'\theta$ and $D\theta$ are variable irreducible wrt. R_S (by lemma 4.10) 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. $y \in \mathcal{X}_g$ for some AC-symbol g in D or D'. Then also $y \in \mathcal{X}_g$ in D_1 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 $tf(v'_1) \cap tf(v'_2)$ do not share any term (and $s'\sigma \neq_{\mathrm{AC}} v'_2$), v'_1 must be a proper subterm of u for some $u =_{\mathrm{AC}} s\sigma$ and v'_2 must be a proper subterm of u' for some $u' =_{\mathrm{AC}} s'\sigma$, which implies v'_1 and v'_2 irreducibles wrt. R_S (as the rules $s\sigma \Rightarrow t\sigma$ and $s'\sigma \Rightarrow t'\sigma$ have been generated). Therefore y is variable irreducible in $D_1\theta$ wrt. R_S.

$D_1\theta$ is an existing instance smaller than C, and moreover variable irreducible. Since S is saturated, the inference must be redundant wrt. R_S, i.e. $R_S \cup AC^{\prec D\theta} \cup Ir_{R_S}(S)^{\prec D\theta} \models D_1\theta$, so from the minimality of C (note that $C \succ D\theta$) we have $I_S \models D_1\theta$. We know $I_S \not\models \Gamma'\sigma, \Gamma\sigma \rightarrow \Delta'\sigma, \Delta\sigma$, 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 $f(f(u_1, u_2), u_3) \rightarrow_{R_S} s_1 =_{\mathrm{AC}\prec C} f(f(t\sigma, v'_1), v_3)$ and $f(u_1, f(u_2, u_3)) \rightarrow_{R_S} s_2 =_{\mathrm{AC}\prec C} f(f(t'\sigma, v'_2), v_3)$ contradict $I_S \not\models C$.

B. If $tf(v_1) = tf(v_2)$ then then $v_1 =_{\mathrm{AC}\prec C} v_2$ and, hence $s\sigma =_{\mathrm{AC}} s'\sigma$. But then, as both $s\sigma \Rightarrow t\sigma$ and $s'\sigma \Rightarrow t\sigma$ have been generated, we have $t\sigma =_{\mathrm{AC}} t'\sigma$, and therefore $s_1 =_{\mathrm{AC}\prec C} s_2$ contradicting the assumption.

C. If $tf(v_1) \subset tf(v_2)$ (the case $tf(v_2) \subset tf(v_1)$ is symmetrical) then let v'_2 and v_3 be ground terms s.t. $v_1 =_{\mathrm{AC}\prec C} v_3$ and $v_2 =_{\mathrm{AC}\prec C} f(v'_2, v_3)$. Therefore $s\sigma =_{\mathrm{AC}} f(s'\sigma, v'_2)$ and the following inference by AC-strict superposition right

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

can be made (with $p' = \lambda$), and, since all the conditions for its application hold, its conclusion D_1 has an instance with a ground substitution θ defined by $\theta = \sigma \cup \{x \mapsto v'_2\}$.

$D_1\theta$ is an existing instance smaller than C, and moreover variable irreducible (as above, since v'_2 is a subterm of some $u =_{\mathrm{AC}} s\sigma$ which implies that it must be irreducible wrt. R_S). Since S is saturated, $R_S \cup AC^{\prec D\theta} \cup Ir_{R_S}(S)^{\prec D\theta} \models D_1\theta$, so from the minimality of C (note that $C \succ D\theta$) we have $I_S \models D_1\theta$. We know $I_S \not\models \Gamma'\sigma, \Gamma\sigma \rightarrow \Delta'\sigma, \Delta\sigma$, so it must be the case that $I_S \models f(t', x)\theta \simeq t\theta$, i.e. $I_S \models f(t'\sigma, v'_2) \simeq t\sigma$, but then $f(f(u_1, u_2), u_3) \rightarrow_{R_S} s_1 =_{\mathrm{AC}\prec C} f(t\sigma, v_3)$ and $f(u_1, f(u_2, u_3)) \rightarrow_{R_S} s_2 =_{\mathrm{AC}\prec C} f(f(t'\sigma, v'_2), v_3)$ 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 $D \ [\![\ T \ ]\!]$ in S, then there are several cases to be analyzed, depending on the maximal equation in C:

1 C has a maximal equation in its succedent.
2 C has a maximal equation $s\sigma \simeq t\sigma$ in its antecedent, with $s\sigma =_{\mathrm{AC}} t\sigma$.
3 C has a maximal equation $s\sigma \simeq t\sigma$ in its antecedent, with $s\sigma \succ t\sigma$.

1. C has a maximal equation $s\sigma \simeq t\sigma$ in its succedent, with $s\sigma \succ t\sigma$: note that $s\sigma \neq_{\mathrm{AC}} t\sigma$, since if C consists only of the equation $s\sigma \simeq t\sigma$ 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; otherwise in fact $s\sigma =_{\mathrm{AC}^{\prec C}} t\sigma$, which, since $I_S \models \mathrm{AC}^{\prec C}$, would contradict $I_S \not\models C$.

Since $I_S \not\models C$, the instance C has not generated the rule $s\sigma \Rightarrow t\sigma$. This must be because conditions 1.b) or 1.c) of definition 4.5 do not hold:

1.1 If condition 1.b) do not hold then D must be $\Gamma \rightarrow \Delta, s' \simeq t', s \simeq t$, with $s\sigma =_{\mathrm{AC}} s'\sigma$ and $I_C \models t\sigma \simeq t'\sigma$. But then the following inference by factoring can be made:

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

Its conclusion D_1 has a ground instance $D_1\sigma$ of the form $\Gamma\sigma, t'\sigma \simeq t\sigma \rightarrow \Delta\sigma, s\sigma \simeq t\sigma$ which is not satisfied by I_S, is smaller than C, and variable irreducible (as C is) wrt. R_S. Since S is saturated, the inference π is redundant in S wrt. R_S. Therefore there exist clauses D'_i in $Ir_{R_S}(S)^{\prec C}$ such that $R_S \cup AC^{\prec C} \cup \{D'_1, \ldots, D'_m\} \models D_1\sigma$. The fact that $D_1\sigma$ is not satisfied by I_S implies that at least one of the D'_i is not satisfied in I_S either, which contradicts the minimality of C.

1.2 If condition 1.c) does not hold then let $s\sigma \simeq t\sigma$ be a strictly maximal equation in $D\sigma$ (note that if $s\sigma \simeq t\sigma$ is only maximal then we are in the previous case) s.t. 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 $maxpos(u, p)$ with $top(u|_{p'}) \neq f$. Note that such a rule always exists. Now there are two main subcases **1.2.1** and **1.2.2**, depending on whether the rule has been generated by an extended instance or by a non-extended instance.

1.2.1 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 $\Gamma' \rightarrow \Delta', s' \simeq t'$, where the rule is $u' \Rightarrow t'\sigma$ for some u' with $s'\sigma =_{\mathrm{AC}} u' = u|_p$.

1.2.1.1 If $maxpos(u, p) = p$ 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 4.3 does not apply either, since $maxpos(s\sigma, p') = p'$ and variables in $\mathcal{X}_f$ for some AC-symbol f always appear below an f. Then the inference by strict superposition right

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

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 $\Gamma'\sigma, \Gamma\sigma \rightarrow \Delta'\sigma, \Delta\sigma, s\sigma[t'\sigma]_{p'} \simeq t\sigma$ that is not satisfied by I_S, is smaller than C, and (as above) variable irreducible wrt. R_S. Since S is saturated, as before, a contradiction is obtained.

1.2.1.2 If $p_1 = maxpos(u,p) \neq p$ 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 below a variable, as before, since again $maxpos(s\sigma, p') = p'$. Then the inference by AC-strict superposition right

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

can be made, and, since all the conditions for its application hold, its conclusion has 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.

1.2.2 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 $maxpos(u,p)$. 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 below a variable, as before, since again $maxpos(s\sigma, p') = p'$, another inference by AC-strict superposition right can be made, whose conclusion, as above, has an instance that contradicts the minimality of C.

2. 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:

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

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.

3. $D\sigma$ 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 4.8 $s\sigma$ 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 1.2: no other rule reduces $s\sigma$ in a position p' below $maxpos(s\sigma, p)$ with $top(u|_{p'}) \neq f$. Now a contradiction is obtained exactly as it is done in case 1.2, but always inferences by (AC-) superposition left are considered instead of (AC-) superposition right. □

DEFINITION 4.14. *Let R be a set of ground rewrite rules. A set of constrained clauses S is* pure *wrt. R if $R \cup Ir_R(S) \cup AC \models S$.*

THEOREM 4.15. *Let S be a saturated set of constrained clauses that is pure wrt. R_S. Then either the empty clause is in S or else $I_S \models S \cup AC$, i.e. S is AC-consistent.*

PROOF. If the empty clause is not in S, then $I_S \models Ir_{R_S}(S) \cup AC$ by Lemma 4.13. Moreover, by construction, $I_S \models R_S$. Therefore, $I_S \models R_S \cup Ir_{R_S}(S) \cup AC$, which, since S is pure wrt. R_S, implies $I_S \models S \cup AC$. □

Now, since we have the previous result for saturated sets, our aim will be to compute such sets, by means of *theorem proving derivations*. The following definition of such derivations is still parameterized by R, so from the definition it is not clear how to compute derivations if R is not known, because e.g. the redundancy of clauses depends on R. Later on this becomes clear, and sufficient conditions for redundancy will be given.

DEFINITION 4.16. *Let R be a ground rewrite system.*

1. *A theorem proving derivation wrt. R is a sequence of sets of constrained clauses $S_0, S_1, \ldots$, where each S_{i+1} is obtained from S_i by adding a logical consequence of $S_i \cup AC$ or by removing a clause $C \,[\![\, T \,]\!]$ that is redundant wrt. R in $S_i \setminus \{C \,[\![\, T \,]\!]\}$.*
2. *The set S_∞ of persistent clauses in $S_0, S_1, \ldots$ is defined as $\cup_j(\cap_{k \geq j} S_k)$.*
3. *A theorem proving derivation wrt. R is* fair *if every inference by* BOAC *with persisting premises is redundant wrt. R in some S_j.*

LEMMA 4.17. *Let $S_0, S_1, \ldots$ be a theorem proving derivation wrt. some R and let π be an inference. If π is redundant (wrt. R) in some S_j then π is redundant wrt. R in S_∞, for $j \geq 0$.*

PROOF. We will prove something more general, namely if π is redundant in $\cup_{j \geq 0} S_j$ then π is redundant in S_∞. Let $\{C_1, \ldots, C_n\}$ be the smallest multiset of ground instances in $Ir_R(\cup_{j \geq 0} S_j)$ s.t. there is an inference π with conclusion D and maximal premise C which is not redundant in S_∞ but $R \cup AC^{\prec C} \cup \{C_1, \ldots, C_n\} \models D$ and $C \succ C_i$ for $i : 1 \ldots n$ (i.e. it is redundant in $\cup_{j \geq 0} S_j$ using $\{C_1, \ldots, C_n\}$). Note that no C_i can be an instance of an AC axiom due to minimality of the multiset. Then there must be some C_i which is a non-persistent instance, i.e. for some k, C_i is an instance of a clause C' in S_k but not of any clause in S_{k+1} and hence C' is redundant in $S_k \setminus \{C'\}$. Therefore, $R \cup Ir_R(S_k)^{\prec C_i} \cup AC^{\prec C_i} \models C_i$, which contradicts the minimality of $\{C_1, \ldots, C_n\}$. □

LEMMA 4.18. *Let $S_0, S_1, \ldots$ be a theorem proving derivation wrt. some R s.t. S_0 is pure wrt. R. Then (i) $R \cup Ir_R(S_\infty) \cup AC \models S_0$ and (ii) S_∞ is pure wrt. R.*

PROOF. (i) We first prove $R \cup Ir_R(S_\infty) \cup AC \models Ir_R(S_0)$, by showing something more general, namely $R \cup Ir_R(S_\infty) \cup AC \models Ir_R(S_k)$ for all k. Let C be the smallest counter example, i.e. C is the minimal clause of all $Ir_R(S_k)$ such that $R \cup Ir_R(S_\infty) \cup AC \not\models C$. Then C is not an instance of any clause in S_∞, so there is some j such that C is an instance of some clause in S_j but not of any clause in S_{j+1}. As in the previous lemma, this means $R \cup Ir_R(S_{j+1})^{\prec C} \cup AC^{\prec C} \models C$, which contradicts the minimality of C.

Now by purity of S_0 (i.e. $R \cup Ir_R(S_0) \cup AC \models S_0$), $R \cup Ir_R(S_\infty) \cup AC \models Ir_R(S_0)$ implies $R \cup Ir_R(S_\infty) \cup AC \models S_0$.

(ii) By definition of theorem proving derivation (only consequences are added) we have $S_0 \cup AC \models S_j$ for $j \geq 0$. Hence $S_0 \cup AC \models (\cup_{j \geq 0} S_j) \supseteq S_\infty$. Therefore, by (i) $R \cup Ir_R(S_\infty) \cup AC \models S_\infty$. □

If we instantiate R by the set R_{S_∞} for some derivation $S_0, S_1, \ldots$, then we get the following theorem. Again, note that R_{S_∞} is not known in advance, and that therefore in practice sufficient conditions for purity and redundancy have to be used.

THEOREM 4.19. *If $S_0, S_1, \ldots$ is a fair theorem proving derivation wrt. R_{S_∞} and S_0 is pure wrt. R_{S_∞}, then (i) S_∞ is saturated, and (ii) either the empty clause is in some S_j (hence S_0 is inconsistent) or else $I_{S_\infty} \models S_j \cup AC$ for $j \geq 0$ (hence S_0 is consistent) and $I_{S_\infty} \models S_\infty \cup AC$.*

PROOF. (i) By fairness of the derivation, we have that for every inference π with premises in S_∞ there is some S_j s.t. π is redundant wrt. R_{S_∞} in S_j. But, by lemma 4.17, π is also redundant wrt. R_{S_∞} in S_∞.

(ii) by lemma 4.13 (applied to S_∞ and R_{S_∞}) and (i), we have that either the empty clause is in S_∞ or else $I_{S_\infty} \models Ir_{R_{S_\infty}}(S_\infty) \cup AC$. Obviously if the empty clause is in S_∞ then it is also in some S_j (and by soundness of the derivation the empty clause is consequence of S_0). Otherwise, by lemma 4.18 (applied to R_{S_∞}), we have $R_{S_\infty} \cup Ir_{R_{S_\infty}}(S_\infty) \cup AC \models S_0$, and hence, as $I_{S_\infty} \models R_{S_\infty}$, it follows that $I_{S_\infty} \models S_0 \cup AC$ (which proves the consistency of S_0). Furthermore, as by soundness of the derivation $S_0 \cup AC \models S_j$ for $j \geq 0$ and $S_0 \cup AC \models S_\infty$, we have $I_{S_\infty} \models S_j \cup AC$ for $j \geq 0$ and $I_{S_\infty} \models S_\infty \cup AC$. □

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. In (Comon *et al.* , 1995) the satisfiability of AC-RPO constraints is proved decidable, but a decision procedure is not crucial here. One only needs sufficiently powerful methods for detecting as many unsatisfiable ordering constraints as possible, which can also be done by e.g. directly applying the AC-RPO ordering on terms with variables.

EXAMPLE 4.20. *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†*. Now an AC-strict superposition right inference by* 1. *on* 2. *with conclusion* 4. $\rightarrow f(c,x) \simeq d \,[\![\, x = f(a,b) \,]\!]$ *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.

† This example is adapted from (Bachmair and Dershowitz, 1989). We thank Leo Bachmair for pointing this example to us and for his comments.

$\rightarrow f(c,y) \simeq d \; [\![\; x = f(a,b) \wedge y = c \;]\!]$. *This is why def. 4.3–3. 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 4.21. *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 (Wertz, 1992), since only* $x \simeq 0 \rightarrow \; [\![\; x = f(0,0) \;]\!]$ *could be obtained.*

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

4.1. Towards more practical simplification and deletion methods

It is not always obvious to decide, in a practical theorem prover, which concrete simplification and deletion techniques to apply, and how to prove their correctness wrt. the abstract redudancy notions of Definition 4.11. An important fact that has to be taken into account in this context is that in this work our main aim was the (theoretical) completeness result of completely avoiding the computation of AC-unifiers. For most practical purposes one would probably choose a reasonable trade-off between this and the need of simplification methods that may require in some cases to instantiate clauses with the unifiers of the constraint part (see below). For example, one might do so for some particular unit clauses that presumably will be useful for many demodulation steps.

In this section we provide some less abstract suffient conditions for checking whether a practical method for detecting redundant clauses or inferences fits into our abstract notions. A more detailed development of such practical methods is a non-trivial tedious task that we feel is beyond the scope of this paper and would require, in combination with results of practical experiments, a deep study on its own. The interested reader can find more details on this subject in Vigneron, 1994, whose practical methods indeed fit into our abstract ones, and in Bachmair *et al.*, 1995 for the non-AC case.

DEFINITION 4.22. *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 4.23. *Let* S *be a set of constrained clauses. A clause* $C \; [\![\; T \;]\!]$ *is redundant in* S *wrt. any rewrite system* R *such that*[†] $\rightarrow_R \subseteq \succ$ *if for every ground instance* $C\sigma$ *of it, there exist ground instances* $D_i\sigma_i$ *in* $S^{\preceq C\sigma}$ *of clauses* $D_i \; [\![\; T_i \;]\!]$, *such that* $\{D_1\sigma_1, \ldots, D_n\sigma_n\} \cup AC^{\preceq C\sigma} \models C\sigma$ *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.

[†] In particular this is the case for R_{S_∞}, which is what is needed in practice.

PROOF. We have to prove that $R \cup Ir_R(S)^{\preceq C\sigma} \cup AC^{\preceq C\sigma} \models C\sigma$ for every set of rules R s.t. $\rightarrow_R \subseteq \succ$ and $C\sigma \in Ir_R(C \llbracket T \rrbracket)$.

We know that $S^{\preceq C} \cup AC^{\preceq C} \models C\sigma$ for all ground instances $C\sigma$ of C, so in particular for all R we have $R \cup S^{\preceq C} \cup AC^{\preceq C} \models C\sigma$ if $C\sigma$ is an irreducible instance of C. It remains to be shown that there are indeed instances in $Ir_R(S)^{\preceq C\sigma}$ that can be used, in case of any of the $D_i\sigma_i$ is not in $Ir_R(S)^{\preceq C\sigma}$.

Now let σ_i' be the ground substitution s.t. $x\sigma_i'$ is the normal form of $x\sigma$ wrt. R if x is not lower bounded by T_i and $x\sigma_i' = x\sigma_i$, otherwise. Since $T_i\sigma_i \equiv true$, $T_i\sigma_i'$ must also be *true*. By the restriction imposed on R we also have $D_i\sigma_i \succeq D_i\sigma_i'$ and obviously $R \cup D_i\sigma_i' \models D_i\sigma_i$, which implies $R \cup \{D_1\sigma_1', \ldots, D_n\sigma_n'\} \cup AC^{\preceq C\sigma} \models C\sigma$.

Now, to prove that these $D_i\sigma_i'$ can indeed be used, i.e. they are in $Ir_R(S)$, we distinguish two cases on the variables x in D_i: (i) if x is not bounded by T_i then $x\sigma_i'$ is in normal form wrt. R and, (ii) if x is bounded by T_i then $x\sigma_i' = y\sigma$ (where y is a variable of C s.t. $y \notin stvars(C, \sigma)$ and there is no AC-symbol f such that y only occurs in C immediately below f), which, by definition, since $C\sigma$ is irreducible, must be irreducible by R. Then, since $x\sigma_i'$ is irreducible by R for every x in $\mathcal{V}ars(D_i)$, obviously $D_i\sigma_i'$ is variable irreducible by R. □

Note that for the "classical" saturation methods without constraint inheritance, the previous lemma shows that the usual redundancy notions are correct, since the first condition is then always true, i.e. a clause C is redundant in S if $S^{\preceq C\sigma} \models C\sigma$ for every ground σ. This means that our framework can deal uniformly with constrained and unconstrained clauses, obtaining exactly the known results for the unconstrained case. In fact, this modified concept of redundant clause (in previous work, $S^{\prec C\sigma} \models C\sigma$ was required) allows us to include practical redundancy methods like subsumption without the need of combining the underlying orderings with special subsumption orderings.

In the constraint case, if both conditions of the previous lemma on the variables fail for some constraint T_i of a clause D_i, and we still want to carry out the redundancy proof, then we can always *weaken* T_i to fulfill the conditions: we can eliminate the part of the constraint that establishes the lower bounds in T_i. Note that it is sound to do this only if we propagate the information of the removed literals in the equality constraint part and we have started with an initial set of clauses without ordering constraints. On the other hand, weakening constraints may increase the number of inferences. Therefore there is sometimes a trade-off between the possibility of carrying out redundancy proofs and restrictedness of the constraints.

A similar lemma holds for the redundancy of inferences:

LEMMA 4.24. *Let S be a set of constrained clauses and let π be an inference (by* BOAC*) with conclusion $C \llbracket T \rrbracket$. The inference π is redundant in S wrt. any rewrite system R if for every ground instance $\pi\sigma$ with maximal premise D, there are ground instances $D_i\sigma_i$ in $S^{\prec D}$ of clauses $D_i \llbracket T_i \rrbracket$, such that $\{D_1\sigma_1, \ldots, D_m\sigma_m\} \cup AC^{\prec D} \models C\sigma$ and for all $i = 1 \ldots m$ and variable x in 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 $y \notin \mathcal{X}_f$ for all AC-symbols f.*

These lemmas provide 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). Cf. Nieuwenhuis and Rubio, 1995 for counter-examples to the completeness in the non-AC case of simplification methods that do not fulfill these conditions.

We do not discuss here which adapted versions for subsumption, simplification, etc. fall into the conditions of the previous lemma. This lemma seems powerful enough for this purpose: we are not aware of any existing correct and complete practical method that cannot be reformulated to follow from it. Let us only remind the reader that it is also possible to weaken (see above, and also Nieuwenhuis and Rubio, 1995) 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 hold.

Finally, let us give a sufficient condition for a constrained clause to have all its ground instances variable reducible wrt. R_{S_∞}, which, according to our redundancy notions, implies that this clause is not needed for computing inferences.

LEMMA 4.25. *Let $S_0, S_1, \ldots$ be a theorem proving derivation wrt. S_∞, with $C \,[\![\, T \,]\!] \in S_j$.*

If for all solutions σ of T, there is some variable x in T such that $x \notin \mathcal{X}_f$ for all AC-symbols f, $x \notin stvars(C, \sigma)$ and $S_j \models x\sigma \simeq s$ for some ground term s with $x\sigma \succ s$ then $Ir_{R_{S_\infty}}(C \,[\![\, T \,]\!]) = \emptyset$.

PROOF. If $S_j \models x\sigma \simeq s$ then $S_\infty \models x\sigma \simeq s$ for all solutions σ of T. Since $I_{S_\infty} \models S_\infty$, we have $I_{S_\infty} \models x\sigma \simeq s$, and by lemma 4.8, since $x\sigma \succ s$, there must be a rule in R_{S_∞} reducing $x\sigma$. Finally as $x \notin \mathcal{X}_f$ and $x \notin stvars(C, \sigma)$, it holds that $C\sigma$ will be variable reducible (wrt. R_{S_∞}) for all ground substitutions σ solution of T. □

In the following example a simple application of this lemma is described:

EXAMPLE 4.26. *Suppose that in some set S_j of the derivation there is a clause like $\rightarrow P(x), Q(x, y) \,[\![\, x = f(a) \,]\!]$ with predicates P and Q, and $S_j \models f(a) \simeq a$. Now from the previous lemma we can conclude that $\rightarrow P(x), Q(x, y) \,[\![\, x = f(a) \,]\!]$ has no irreducible (wrt. R_{S_∞}) instances and therefore, that it is redundant and can be removed without losing completeness.*

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 $\rightarrow_{E\setminus\mathrm{AC}}$. 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 $s[u] \rightarrow_{E\setminus\mathrm{AC}} s[r\sigma]$ if $u =_{\mathrm{AC}} l\sigma$ and $s[u] \rightarrow_{E\setminus\mathrm{AC}} s[f(r, x)\sigma]$ 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*
$E \cup AC \models s \simeq t$ *iff* $s \to^*_{E/\mathrm{AC}} \circ \leftrightarrow^*_{\mathrm{AC}} \circ \leftarrow^*_{E/\mathrm{AC}} t$ *iff* $s \to^*_{E\setminus\mathrm{AC}} \circ \leftrightarrow^*_{\mathrm{AC}} \circ \leftarrow^*_{E\setminus\mathrm{AC}} t$.

PROOF. We prove that every ground term u (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). Then the theorem holds since $E \models s \simeq t$ iff the Skolemized versions of s and t have the same minimal (up to AC-equality) representative. We proceed by induction wrt. $\succ$, i.e. it suffices to prove the reducibility wrt. $\to_{E/\mathrm{AC}}$ and $\to_{E\setminus\mathrm{AC}}$ of non-minimal s.

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

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 fulfills the equality constraint part T' of T (the ordering part can be ignored). Otherwise, it suffices to instantiate the extra variables with the adequate most general unifier 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 $\Gamma \to \;[\![\, T \,]\!]$ (which need not be used) and Horn clauses $\Gamma \to s \simeq t \;[\![\, T \,]\!]$ 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 $\mathcal{V}ars(s\theta) \supseteq \mathcal{V}ars(\Gamma\theta)$.

6. Conclusions and Further Work

Symbolic constraints turn out to be especially useful in the context of built-in equational theories E (apart from the reduced search space due to the inherited ordering and unification restrictions of the inferences, cf. Nieuwenhuis and Rubio, 1995). In every inference, instead of producing as many conclusions as minimal E-unifiers of two terms s and t, one single conclusion is generated with an additional E-unification problem $s = t$ kept in its constraint.

In the AC-case developed here (AC is the most extensively studied built-in theory), the computation of the —doubly exponentially many— AC-unifiers can thus be completely avoided. For dealing with the AC properties, a new simpler proof technique has been defined, which also covers uniformly —and improves upon— the existing results for rewriting and Knuth-Bendix completion modulo AC.

Of course it would be interesting to develop similar techniques for other "well-behaved"

equational theories, where the strongest requirement for being well-behaved seems to be the existence of an E-compatible total simplification ordering, rather than the finiteness of minimal complete sets of E-unifiers. Note that the latter property is not required any more with the ideas developed in this paper, provided that an adequate inference system and an ordering can be found, and similar completeness methods can be applied. These ideas have been developed in (Rubio, 1995) for the case where the equational theory consists of a set of associativity (A) axioms for a subset of the function symbols; indeed A-unification is infinitary.

In fact, even the decidability of the E-unification problem is not essential any more; when an empty clause with a constraint T is generated, one can semi-decide the satisfiability of T by an independent prover in parallel with the theorem proving procedure.

As another conclusion of this work, it seems that research on AC-, and in general E-unification should focus more on efficiently attacking the decision problem, rather than on computing complete sets of unifiers. The development of efficient (possibly incomplete) tests for E-unifiability should also take into account the need of *incremental* algorithms in this context: as in constraint logic programming, when adding new restrictions to an existing constraint T, one should be able to re-use the work done for T.

References

Baader, F., Siekmann, J. (1993). 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).

Bachmair, L., Dershowitz, N. (1987). Completion for rewriting modulo a congruence. In Pierre Lescanne, editor, *Rewriting Techniques and Applications, [2nd International Conference]*, LNCS 256, pages 192–203, Bordeaux, France, May 25–27. Springer-Verlag.

Bachmair, L., Dershowitz, N. (1989). Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 2 and 3(67):173–201.

Bachmair, L., Ganzinger, H. (1994). Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247.

Bachmair, L., Ganzinger, H., Lynch, C., Snyder, W. (1992). 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. Springer-Verlag.

Bachmair, L., Ganzinger, H., Lynch, C., Snyder, W. (1995). Basic paramodulation. *Information and Computation*, 121(2):172–192.

Bachmair, L. (1991). *Canonical equational proofs*. Birkhäuser, Boston, Mass..

Boudet, A., Contejean, E., Devie, H. (1990). A new AC unification algorithm with an algorithm for solving systems of diophantine equations. In *Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 289–299, Philadelphia, Pennsylvania, USA, June 4–7. IEEE Computer Society Press, Los Alamitos, CA, USA.

Bürckert, H-J., Herold, A., Kapur, D., Siekmann, J.H., Stickel, M.E., Tepp, M., Zhang, H. (1988). Opening the AC-Unification Race. *Journal of Automated Reasoning*, 4(1):465–474.

Comon, H., Nieuwenhuis, R., Rubio, A. (1995). Orderings, AC-Theories and Symbolic Constraint Solving. In *Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 375–385, San Diego, California, USA, June 26–29. IEEE Computer Society Press, Los Alamitos, CA, USA.

Dershowitz, N., Jouannaud, J-P. (1990). Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 6, pages 244–320. Elsevier Science Publishers B.V., Amsterdam, New York, Oxford, Tokyo.

Domenjoud, E. (1992). AC unification through order-sorted AC1 unification. *Journal of Symbolic Computation*, 14(6):537–556.

Domenjoud, E. (1992). 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.

Fages, F. (1987). Associative-commutative unification. *Journal of Symbolic Computation*, 3(3).

Fortenbacher, A. (1987). An algebraic approach to unification under associativity and commutativity. *Journal of Symbolic Computation*, 3.

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

Huet, G. (1980). Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821.

Jouannaud, J-P., Kirchner, H. (1986). Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15:1155–1194.

Jouannaud, J-P. (1983). Confluent and coherent equational term rewriting systems: Applications to proofs in abstract data types. In *Proc. 8th Colloquium on Trees in Algebra and Programming*, LNCS 59, pages 269–283. Springer-Verlag.

Kapur, D., Narendran, P. (1992). Complexity of unification problems with associative commutative operators. *Journal of Automated Reasoning*, 9:261–288.

Kapur, D., Narendran, P. (1992). Double exponential complexity of computing complete sets of AC-unifiers. In *Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 11–21, Santa Cruz, California, USA. IEEE Computer Society Press.

Kirchner, C., Kirchner H., Rusinowitch, M. (1990). Deduction with symbolic constraints. *Revue Française d'Intelligence Artificielle*, 4(3):9–52.

Kirchner, C. (1989). From unification in combination of equational theories to a new AC-unification algorithm. In Hassan Aït-Kaci and Maurice Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2: Rewriting Techniques, chapter 6, pages 171–210. Academic Press.

Lankford, D.S., Ballantyne, A.M. (1977). Decision procedures for simple equational theories with commutative-associative axioms: Complete sets of commutative-associative reductions. Technical Report Memo ATP-39, Dept. of Mathematics and Computer Science, Univ. of Texas, Austin, TX.

Lincoln, P., Christian, T. (1989). Adventures in associative-commutative unification. *Journal of Symbolic Computation*, 8(1–2):217–240.

Narendran, P., Rusinowitch, M. (1991). 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. Springer-Verlag.

Nieuwenhuis, R., Rubio, A. (1992). Basic superposition is complete. In B. Krieg-Brückner, editor, *European Symposium on Programming*, LNCS 582, pages 371–390, Rennes, France. Springer-Verlag.

Nieuwenhuis, R., Rubio, A. (1995). Theorem Proving with Ordering and Equality Constrained Clauses. *J. of Symbolic Computation*, 19(4):321–351.

Pais, J., Peterson, G. (1991). Using Forcing to Prove Completeness of Resolution and Paramodulation. *Journal of Symbolic Computation*, 11(1):3–19.

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

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

Robinson, G.A., Wos, L.T. (1969). Paramodulation and theorem-proving in first order theories with equality. *Machine Intelligence*, 4:135–150.

Rubio, A., Nieuwenhuis, R. (1995). A total AC-compatible ordering based on RPO. *Theoretical Computer Science*, 142(2):209–227.

Rubio, A. (1994). Automated deduction with ordering and equality constrained clauses. PhD. Thesis, Technical University of Catalonia, Barcelona, Spain.

Rubio, A. (1995). Theorem Proving modulo Associativity. In *1995 Conf. of the European Association for Computer Science Logic*, LNCS, Paderborn, Germany. Springer-Verlag.

Rusinowitch, M., Vigneron, L. (1993). Automated deduction with associative commutative operators. Rapport de Recherche 1896, Institut National de Recherche en Informatique et en Automatique, INRIA-Lorraine. To appear in J. of Applicable Algebra in Engineering, Communication and Computation.

Stickel, M.E. (1981). A unification algorithm for associative-commutative functions. *Journal of the ACM*, 28(3):423–434.

Vigneron, L. (1994). Associative Commutative Deduction with constraints. In Allan Bundy, editor, *12th International Conference on Automated Deduction*, LNAI 814, pages 530–544, Nancy, France. Springer-Verlag.

Wertz, U. (1992). First-order theorem proving modulo equations. Technical Report MPI-I-92-216, Max-Planck-Institut für Informatik, Saarbrücken.