Theorem Proving modulo Associativity*

Albert Rubio

Universitat Politècnica de Catalunya,
Pau Gargallo 5, 08028 Barcelona, Spain
E-mail: rubio@lsi.upc.es. Tel. 34-3-4017330

Abstract. We present an inference system for first-order constrained clauses with equality modulo associativity (A). Our procedure is refutationally complete and reduces to Knuth-Bendix completion modulo A in the equational case. As an essential ingredient we present the first —as far as we know— A-compatible reduction ordering total on the ground A-congruence classes.

1 Introduction

In some cases, in automated theorem proving special treatments for some equational subset of the axioms are useful. 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 two other good reasons for applying specific methods for some particular equations. On one hand, some axioms generate many slightly different permuted versions of clauses, and for efficiency reasons it is many times better to treat all these clauses together as a single one representing the whole class. In the case modulo associativity (A), like it also happens in the case modulo associativity and commutativity (AC), there is a second advantage: sets like $f(a, b) \to b$ and $f(a, f(x, b)) \to f(x, b)$ are complete modulo A, but cannot be completed without building-in associativity of f [14].

Also, as observed by Bachmair in [1], 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). This leads to extending completion procedures 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 inferences with special E-matching and E-unification algorithms [8, 5, 14, 6, 7, 2], some of them for specific sets E and for wider classes of sets others, but all of them requiring that a complete E-unification algorithm is given.

In general there is no unique most general E-unifier for a given E-unification problem. In the case of A-unification, unlike in AC-unification, the *minimal complete* set of unifiers needs not even to be finite: $f(a, x)$ and $f(x, a)$ have an infinite set of independent A-unifiers: $a, f(a, a), f(a, a, a), \ldots$ Due to this, equality constraints [11, 12, 13] are extremely useful in this context: instead of E-unifying the terms, the unification problem is kept in an (E-)equality constraint. Dealing with a constrained clause $C \llbracket\ s = t\ \rrbracket$ can be much more efficient than having n clauses $C_1, \ldots, C_n$, one for each E-unifier that is a solution of $s = t$, since e.g. many inferences can be computed at once, and each inference generates *one single conclusion* with an additional equality $s = t$ in its constraint. Furthermore, computing A-unifiers is not needed at all, which is of course crucial, due its well-known computational and algorithmic complexity [9]. A clause C with an A-equality constraint T 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 decide the A-unifiability of T to know whether an inconsistency has been derived or not. By means of the model generation

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

method [3], we proved the completeness of such a basic strategy for the AC-case in [12] (see also [17]), combined with ordering constraints. Here we develop a similar inference system for the case modulo A, together with an essential ingredient for these methods, namely the first —as far as we know— A-compatible reduction ordering total on the ground A-congruence classes.

This paper is structured as follows. After the introduction some basic notions and definitions are given. In section 3 the A-compatible ordering is defined. Section 4 is devoted to the inference system and section 5 to prove its refutationally completeness. In section 6 the usefulness of saturated sets is studied. All proofs and detailed explanations of the following can be found in [15].

2 Preliminaries

Below let s and t (possibly indexed) be terms in $\mathcal{T}(\mathcal{F})$ and f a function symbol in $\mathcal{F}$ and let E be an equational theory. A (strict partial) ordering on $\mathcal{T}(\mathcal{F})$ (a transitive irreflexive relation) $\succ$ fulfills the *subterm property* if $f(\ldots t \ldots) \succ t$ for all f and t. It is *monotonic* if $s \succ t$ implies $f(\ldots s \ldots) \succ f(\ldots t \ldots)$. A monotonic ordering fulfilling the subterm property is a *simplification ordering* and for finite signatures it is *well-founded*: there is no infinite sequence $t_1 \succ t_2 \succ \ldots$ An ordering $\succ$ is *E-compatible* if $s' =_E s \succ t =_E t'$ implies $s' \succ t'$. Finally we say that an ordering is E-total on $\mathcal{T}(\mathcal{F})$ if $s \neq_E t$ implies $s \succ t$ or $t \succ s$ for all terms $s, t \in \mathcal{T}(\mathcal{F})$. Any well-founded monotonic ordering $\succ$ total on $\mathcal{T}(\mathcal{F})$ is a simplification ordering.

An ordering on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is *stable* under substitutions if $s \succ t$ implies $s\sigma \succ t\sigma$ for all terms s and t and substitutions σ. A well-founded monotonic ordering stable under substitutions is called a *reduction ordering*.

Here an *equation* is a multiset of terms $\{s, t\}$, which will be written in the form $s \simeq t$. 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, in what follows, $=$ will be interpreted as A-equality $=_A$, and $\succ$ as an A-compatible simplification ordering A-total on ground terms. 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 antecedent get a slightly higher complexity than in the succedent. An occurrence of an equation $t \simeq t'$ in an antecedent is (the two-fold multiset) $\{\{t, t'\}\}$, and in a succedent it is $\{\{t\}, \{t'\}\}$. Now the two-fold multiset extension of $\succ$ is A-total on ground equations, and the three-fold multiset extension of $\succ$ is an A-total ordering on ground clauses. We will ambiguously use $\succ$ to denote all these orderings on 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 $e' =_{\mathrm{AC}} e$).

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).

An (A-) interpretation I is a congruence on ground terms (satisfying the A-axiom for all A-operators). It satisfies a ground clause $\Gamma \rightarrow \Delta$, denoted $I \models \Gamma \rightarrow \Delta$, if $I \not\supseteq \Gamma$ or else $I \cap \Delta \neq \emptyset$. An interpretation I satisfies (is a model of) a constrained clause $C \llbracket T \rrbracket$, denoted $I \models C \llbracket T \rrbracket$, if it satisfies every ground instance of $C \llbracket T \rrbracket$, i.e. every $C\sigma$ such that σ is ground and $T\sigma$ is true. Therefore, clauses with unsatisfiable constraints are tautologies, and $C \llbracket T \rrbracket$ 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 $\succ$).

We use the definitions of [4] 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 A-operators f that are immediately below another f. For example, if f and g are A-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 A-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 A-equal terms, i.e. $s =_{\mathrm{AC}} t$. Then their flattened forms are equal: 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 A-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 A total A-compatible reduction ordering

As for the C-case, it is clear that AC-compatible orderings are also A-compatible orderings (when the associative symbols are also considered commutative), but what is not so clear is how to get an A-compatible reduction ordering A-total on $\mathcal{T}(\mathcal{F})$. Suppose we are trying to extend a path ordering. Then the first attempt could be to flatten terms to ensure A-compatibility and then apply a *lexicographic path ordering* (LPO; see e.g. [4]) to ensure totality. But due to flattening, we have variable arity function symbols (the A-symbols), and this leads to loose well-foundedness, since LPO is a simplification ordering only when all symbols have fixed arity (as it does not fulfil the deletion property): if $a \succ b$ then $f(b,a) \succ f(b,b,a) \succ f(b,b,b,a) \succ \ldots$. Another attempt could be to normalize terms wrt. associativity rules (moving A-symbols to the right or to the left) and then apply LPO (on terms with fixed-arity function symbols), but again monotonicity is lost: suppose f associative and $f \succ g$, and that we normalize wrt. the rule $f(f(x,y),z) \to f(x,f(y,z))$ (i.e. we move A-symbols to the right); then $f(a,a) \succ_A g(a)$ but $f(g(a),a) \succ_A f(f(a,a),a)$.

In the following we will define an A-compatible reduction (simplification) ordering $\succ_A$ extending an AC-compatible reduction (simplification) ordering $\succ_{AC}$ s.t. $\succ_A$ is A-total on $\mathcal{T}(\mathcal{F})$ whenever $\succ_{AC}$ is AC-total on $\mathcal{T}(\mathcal{F})$, provided that $\mathcal{F}$ is a set of fixed-arity function symbols and $\mathcal{F}_A$ the subset of associative function symbols (cosidered associative-commutative when using $\succ_{AC}$).

Definition 1. Let $\mathcal{F}$ be a set of fixed-arity function symbols, $\succ_{AC}$ an ordering on $\mathcal{T}(\mathcal{F},\mathcal{X})$ and s and t terms in $\mathcal{T}(\mathcal{F},\mathcal{X})$, with $tf(s) = \langle s_1, \ldots, s_m\rangle$ and $tf(t) = \langle t_1, \ldots, t_n\rangle$ (here we have flattening rules for all symbols in $\mathcal{F}_A$, but since they are only applied at top-most position all terms in the top-flattening are in $\mathcal{T}(\mathcal{F},\mathcal{X})$, i.e. there is no problem with the requirement of having fixed-arity function symbols). Then $s \succ_A t$ iff

$$s \succ_{AC} t \quad \text{or} \quad s =_{\mathrm{AC}} t \quad \text{and} \quad \langle s_1, \ldots, s_m\rangle \succ_A^{lex} \langle t_1, \ldots, t_n\rangle$$

where $\succ_A^{lex}$ is the lexicographic extension of $\succ_A$ wrt. $=_A$ (note that if $s =_{\mathrm{AC}} t$ then $top(s) = top(t)$ and $m = n$), and $tf(u)$ for some term u headed by f denotes the list of subterms of u at maximal position not headed by f if f is in $\mathcal{F}_A$, and the list of the arguments of u otherwise.

Theorem 2. *Let $\succ_{AC}$ be an AC-compatible reduction (simplification) ordering on terms in $\mathcal{T}(\mathcal{F},\mathcal{X})$. Then $\succ_A$ is an A-compatible reduction (simplification) ordering on $\mathcal{T}(\mathcal{F},\mathcal{X})$. Moreover, if $\succ_{AC}$ is AC-total on $\mathcal{T}(\mathcal{F})$ then $\succ_A$ is A-total on $\mathcal{T}(\mathcal{F})$.*

Proof. All properties are proved separately by induction on the size of the involved terms. In the following suppose $tf(s) = \langle s_1, \ldots, s_m\rangle$ and $tf(t) = \langle t_1, \ldots, t_n\rangle$.

1. Irreflexivity and transitivity follow from the irreflexivity and transitivity of $\succ_{AC}$ and $\succ_A^{lex}$ (by induction hypothesis).
2. Monotonicity. Suppose $s \succ_A t$: If $s \succ_{AC} t$ then $f(\ldots s \ldots) \succ_{AC} f(\ldots t \ldots)$ (by monotonicity of $\succ_{AC}$) and hence $f(\ldots s \ldots) \succ_A f(\ldots t \ldots)$. If $s =_{\mathrm{AC}} t$ then $top(s) = top(t)$, $m = n$, $tf(s) \succ_A^{lex} tf(t)$ and $f(\ldots s \ldots) =_{\mathrm{AC}} f(\ldots t \ldots)$. There are two possible cases: if $top(s) = f$ is an A-symbol then $tf(f(\ldots s \ldots)) = \langle \ldots, s_1, \ldots, s_m \ldots\rangle \succ_A^{lex} \langle \ldots, t_1, \ldots, t_n \ldots\rangle = tf(f(\ldots t \ldots))$. Otherwise, we have $tf(f(\ldots s \ldots)) = \langle \ldots, s \ldots\rangle \succ_A^{lex} \langle \ldots, t \ldots\rangle = tf(f(\ldots t \ldots))$.
3. Well-foundedness. Suppose there exists an infinite decreasing sequence $u_1 \succ_A u_2 \succ_A u_3 \succ_A \ldots$. Since $\succ_{AC}$ is well-founded and by transitivity of $\succ_A$, there exists k s.t. $u_k =_{\mathrm{AC}} u_j$ for all $j \geq k$. But there is only a finite number of (syntactically different) terms AC-equivalent to u_k, which implies that there must be two terms u_q and u_p with $p > q \geq k$ s.t. u_q and u_p are the same term (i.e. they are syntactically equal). Therefore by transitivity we have $u_q \succ_A u_p$ which contradicts the irreflexivity.
4. Stability under substitutions (when dealing with terms with variables). Suppose $s \succ_A t$. If $s \succ_{AC} t$ then by stability of $\succ_{AC}$ we have $s\sigma \succ_{AC} t\sigma$, and hence $s\sigma \succ_A t\sigma$, for all ground substitutions σ. If $s =_{\mathrm{AC}} t$ then $s\sigma =_{\mathrm{AC}} t\sigma$ for all ground substitutions σ and $\langle s_1, \ldots, s_m\rangle \succ_A{}^{lex} \langle t_1, \ldots, t_n\rangle$, i.e. $s_1 =_A t_1 \wedge \ldots \wedge s_{i-1} =_A t_{i-1} \wedge s_i \succ_A t_i$, which implies by induction hypothesis $s_1\sigma =_A t_1\sigma \wedge \ldots \wedge s_{i-1}\sigma =_A t_{i-1}\sigma$ and $s_i\sigma \succ_A t_i\sigma$. Now suppose $tf(s\sigma) = \langle s'_1, \ldots, s'_p\rangle$ and $tf(t\sigma) = \langle t'_1, \ldots, t'_p\rangle$ (note that due to the existence of variables $tf(s\sigma)$ could be different from $tf(s)\sigma$ as well as for t). Since s_i can not be a variable (otherwise it would not be greater than any term), there must be some $j = 1 \ldots p$ s.t. s'_j is $s_i\sigma$ and therefore $s'_1 =_A t'_1 \wedge \ldots \wedge s'_{j-1} =_A t'_{j-1} \wedge s'_j \succ_A t'_j$ (since s'_j is $s_i\sigma$ and t'_j is $t_i\sigma$ or one of its subterms), which implies $s\sigma \succ_A t\sigma$.
5. Subterm property (note that the deletion property trivially holds since all symbols are supposed to have fixed arity). Since $\succ_{AC}$ satisfies this property we have $f(\ldots s \ldots) \succ_{AC} s$ and hence $f(\ldots s \ldots) \succ_A s$.
6. A-compatibility. Suppose we have $s' =_A s \succ_A t =_A t'$. If $s \succ_{AC} t$, since $s' =_A s$ and $t' =_A t$ implies $s' =_{\mathrm{AC}} s$ and $t' =_{\mathrm{AC}} t$, by AC-compatibility of $\succ_{AC}$ we have $s' \succ_{AC} t'$ and hence $s' \succ_A t'$. If $s =_{\mathrm{AC}} t$ then $s' =_{\mathrm{AC}} t'$ and $tf(s) \succ_A{}^{lpo} tf(t)$. Suppose $tf(s') = \langle s'_1, \ldots, s'_m\rangle$ and $tf(t') = \langle t'_1, \ldots, t'_n\rangle$. Therefore $s_i =_A s'_i$ and $t_i =_A t'_i$ for all $i = 1 \ldots m$ (note that $m = n$) and by induction hypothesis $tf(s') \succ_A{}^{lpo} tf(t')$, which implies $s' \succ_A t'$.
7. A-totality. By AC-totality of $\succ_{AC}$ we know that either $s \succ_{AC} t$ or $t \succ_{AC} s$ or $s =_{\mathrm{AC}} t$ for all ground terms s and t. Therefore, either $s \succ_A t$ or $t \succ_A s$ or we have to compare $\langle s_1, \ldots, s_m\rangle$ and $\langle t_1, \ldots, t_n\rangle$. But, by induction hypothesis for all $i = 1 \ldots m$, either $s_i \succ_A t_i$ or $t_i \succ_A s_i$ or $s_i =_A t_i$, and hence either $s =_A t$ or $tf(s) \succ_A{}^{lpo} tf(t)$ implying $s \succ_A t$ or $tf(t) \succ_A{}^{lpo} tf(s)$ implying $t \succ_A s$. □

As seen in the proof, the only restriction needed is the fact that all function symbols have fixed arity. Terms compared must be built fulfilling this condition, but we do not mind about how these terms are compared under $\succ_{AC}$, i.e. if terms are flattened or not, or if $\succ_{AC}$ is based on polynomial interpretation or on a path ordering, etc. Therefore, there are no other restrictions on the ordering $\succ_{AC}$ but to be an AC-compatible reduction (simplification) ordering (possibly AC-total on $\mathcal{T}(\mathcal{F})$). Since such an AC-total ordering for the AC-case indeed exists (e.g. [10, 16]), from the previous theorem the existence of an A-total A-compatible reduction ordering follows.

4 The inference system I_A

The following inference system called I_A extends the classical strict superposition inference system I given in e.g. [13].

Definition 3. In the inference rules the ordering constraints (OC) always encode that the equations $s \simeq t$ and $s' \simeq t'$ are maximal wrt. $\succ$ (which is an A-compatible reduction ordering total on the A-congruence classes) in the premise to which they belong (strictly maximal if they belong to the succedent), with $s \succ t$ and $s' \succ t'$, and that $s \simeq t$ is bigger than $s' \simeq t'$ except in A-top superposition.

In the A-superposition rules the variables x and x' (which are new variables) can be empty. Furthermore if x or x' are not empty then the term s' can be restricted to be headed by the A-symbol f. This can be expressed in the constraint language and added to the constraint. Similarly to the AC-case, A-top superposition is only needed if s and s' are headed by f and the top-flattening (i.e. applying flattening only at top-most position) of x is a proper prefix of the top-flattening of s' and the top-flattening of x' is a proper suffix of the top-flattening of s. Note that here we cannot restrict the inference to consider only the longest shared prefix and suffix respectively of s and s' (see example 1).

Of course, the superposition inferences are needed only if $s|_p$ is non-variable, and A-superposition is needed only if moreover $s|_p$ (which has an f as top symbol) is not immediately below another f.

1. *A-strict superposition right*:

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

2. *A-strict superposition left*:

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

3. *A-top-superposition*:

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

4. *equality resolution:*

$$\frac{\Gamma, s \simeq t \to \Delta \llbracket T \rrbracket}{\Gamma \to \Delta \llbracket T \wedge OC \wedge\ s = t \rrbracket}$$

5. *factoring:*

$$\frac{\Gamma \to \Delta, s' \simeq t', s \simeq t \llbracket T \rrbracket}{\Gamma, t \simeq t' \to \Delta, s \simeq t \llbracket T \wedge OC \wedge\ s = s' \rrbracket}$$

In the rules of A-strict superposition right and left the fact that x or x' can be empty means that apart from $f(x, f(s', x'))$ we have to consider in addition $f(x, s')$, $f(s', x')$ and s' to superpose with.

Example 1. For simplicity in this example we use string notation (but there is no λ) for the terms. Suppose a simplification ordering on strings extending $a \succ b \succ c \succ d \succ e$. Then the following set:

1. $\to baa \simeq d$
2. $\to aac \simeq e$
3. $\to be \simeq dc$
4. $bae \simeq dac \to$

is inconsistent, but due to the ordering restrictions (note that $baa \succ d$ and $aac \succ e$ by the transitivity and the subterm property), the only one possible inference is by A-top superposition between 1. and 2. but, instead of taking the longest shared part (getting 3.), considering just one a symbol to be shared, which produces 5. $\to bae \simeq dac$. Finally 5. and 4. lead to the empty clause.

5 Refutation completeness

Definition 4. 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 $\Gamma \rightarrow \Delta, s \simeq t$ 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 terms v and v', the clauses (i) $\Gamma \rightarrow \Delta, f(s,v) \simeq f(t,v)$ (ii) $\Gamma \rightarrow \Delta, f(v,s) \simeq f(v,t)$ and (iii) $\Gamma \rightarrow \Delta, f(v, f(s,v')) \simeq f(v, f(t,v'))$ are called *extended* (ground) instances of $C \llbracket T \rrbracket$ wrt. $C\sigma$ (with the *context* v and 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 I_A) with premises $C_1 \llbracket T_1 \rrbracket, \ldots C_n \llbracket T_n \rrbracket$ and conclusion $C \llbracket T \rrbracket$ 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 5. 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 6. A variable x is called an *extension variable* for the A-symbol f if it is the new variable x introduced in an inference by A-strict superposition (right or left). The set of all extension variables for the function symbol f is denoted by $\mathcal{X}_f$.

Definition 7. 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 A-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.

Definition 8. Let R be a set of ground rewrite rules, $C \llbracket T \rrbracket$ a constrained clause, S a set of constrained clauses and π an inference.

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

Definition 9. Let $D \llbracket T \rrbracket$ be a constrained clause and let π be an inference.

1. $D \llbracket T \rrbracket$ is *redundant* in a set of constrained clauses S if $Ir_R(S)^{\preceq C} \cup A^{\preceq C} \cup R \models C$
2. π is *redundant* in a set S if $Ir_R(S)^{\prec C} \cup A^{\prec C} \cup R \models D$

for every set of rules R s.t. $\rightarrow_R \subseteq \succ$ and C in $Ir_R(D \llbracket T \rrbracket)$, and for every inference in $Ir_R(\pi)$ with maximal premise C and conclusion D.

At the end of this section we will provide some sufficient conditions for proving redundancy which include well-known *practical* methods like subsumption or rewriting.

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 10. 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 A_C denote the set of ground instances $s \simeq s'$ of equations in A with $C \succ s \simeq s'$ and s and s' irreducible wrt. R_C, and let I_C denote the interpretation $(R_C \cup A_C)^*$.

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 all rules $u \Rightarrow t$ with $u =_A s$ if:
   (a) $I_C \not\models C$,
   (b) $I_C \not\models t \simeq t'$, for every $s' \simeq t'$ in Δ with $s' =_A s$, and
   (c) u is irreducible by R_C for all u with $u =_A s$.
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 $u =_A s$ and u irreducible by R_C, C generates a rule $u \Rightarrow t$.

Finally[2], we define $R_S = \bigcup R_C$, $A_S = \bigcup A_C$, and $I_S = (R_S \cup A_S)^*$.

Note that since $\succ$ is A-compatible $s \succ t$ implies $u \succ t$ for all u with $u =_A s$.

Lemma 11. *Let S be a set of constrained clauses, and C and D instances of clauses in S with $C \succ D$. Then $R_C \supseteq R_D$, $A_C \supseteq A_D$, and $I_C \supseteq I_D$.*

Lemma 12. *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 $l =_A l'$ then r and r' are the same term.*
4. *There are no overlaps between left hand sides of rules of R_S.*
5. *If $u =_A v$ for ground terms u and v, then u is reducible by R_S iff v is.*

Lemma 13. *Let S be a set of constrained clauses, and let s and t be ground terms. Then $I_S \models s \simeq t$ implies $s \rightarrow^*_{R_S} s' =_{A_S} t' \leftarrow^*_{R_S} t$ for some s' and t'.*

Lemma 14. *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 $u =_{AC} s\sigma$ then $I_S \supseteq \Gamma\sigma$ and $I_S \cap \Delta\sigma = \emptyset$.*

Lemma 15. *Let S be a set of constrained clauses. If a ground instance $C\sigma$ of a clause $C [\![T]\!]$ in S, with C of the form $\Gamma \rightarrow \Delta, s \simeq t$ generates rules $u \Rightarrow t\sigma$ with $u =_A s\sigma$ then $C\sigma$ is variable irreducible wrt. R_S.*

Definition 16. A set of constrained clauses S is *saturated* if all inferences by I_A with premises in S are redundant in S.

Lemma 17. *Let S be a saturated set of constrained clauses not containing the empty clause. Then $I_S \models Ir_{R_S}(S) \cup A$.*

[2] to ensure that in this inductive definition all instances of A 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 $\rightarrow x \simeq x [\![\mathit{true}]\!]$ is in S (note that this clause cannot generate any rule).

Proof. We will derive a contradiction from the existence of a minimal (wrt. $\succ$) instance C in $Ir_{R_S}(S) \cup A$ such that $I_S \not\models C$.

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 $C \notin A_S$, 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 by lemma 12 point 5 both of them, since they are A-equal. Consider u_1, u_2 and u_3 irreducible (otherwise a contradiction is directly obtained).

Let the rule reducing $f(f(u_1, u_2), u_3)$ be $l \Rightarrow r$ with (i) $l =_A s\sigma$ and $r = t\sigma$, (ii) $l =_A f(s\sigma, v)$, and $r = f(t\sigma, v)$ (iii) $l =_A f(v, s\sigma)$ and $r = f(v, t\sigma)$ or (iv) $l =_A f(v, f(s\sigma, w))$ and $r = f(v, f(t\sigma, w))$, generated by an instance with σ (possibly extended with the context v (and w)) of a clause D of the form $\Gamma \to \Delta, s \simeq t$, where $top(s\sigma) = f$. Then $f(f(u_1, u_2), u_3) =_A f(s\sigma, v_1)$ or $f(f(u_1, u_2), u_3) =_A f(v_2, s\sigma)$ or $f(f(u_1, u_2), u_3) =_A f(v_3, f(s\sigma, v_4))$ for some v_1, v_2, v_3 and v_4; and $f(f(u_1, u_2), u_3)$ is rewritten into some s_1 with $s_1 =_{A^{\prec C}} f(t\sigma, v_1)$ or $s_1 =_{A^{\prec C}} f(v_2, t\sigma)$ or $s_1 =_{A^{\prec C}} f(v_3, f(t\sigma, v_4))$ (we use $A^{\prec C}$ to denote the instances of A that are smaller wrt. $\succ$ than C; note that $I_S \models A^{\prec C}$ by minimality of C).

Similarly, let the rule reducing $f(u_1, f(u_2, u_3))$ be $l' \Rightarrow r'$ with (i) $l' =_A s'\sigma$ and $r' = t'\sigma$, (ii) $l' =_A f(s'\sigma, v')$, and $r' = f(t'\sigma, v')$ (iii) $l' =_A f(v', s'\sigma)$ and $r' = f(v', t'\sigma)$ or (iv) $l' =_A f(v', f(s'\sigma, w'))$ and $r' = f(v', f(t'\sigma, w'))$, generated by an instance with σ (possibly extended with the context v' (and w')) of a clause D' of the form $\Gamma' \to \Delta', s' \simeq t'$, where $top(s'\sigma) = f$. Then $f(f(u_1, u_2), u_3) =_A f(s'\sigma, v'_1)$ or $f(f(u_1, u_2), u_3) =_A f(v'_2, s'\sigma)$ or $f(f(u_1, u_2), u_3) =_A f(v'_3, f(s'\sigma, v'_4))$ for some v'_1, v'_2, v'_3 and v'_4; and $f(f(u_1, u_2), u_3)$ is rewritten into some s_2 with $s_2 =_{A^{\prec C}} f(t'\sigma, v'_1)$ or $s_2 =_{A^{\prec C}} f(v'_2, t'\sigma)$ or $s_2 =_{A^{\prec C}} f(v'_3, f(t'\sigma, v'_4))$.

It can be assumed wlog. that $s\sigma \succ s'\sigma$ or $s\sigma =_A s'\sigma$. Let $tf(f(f(u_1, u_2), u_3))$ (and hence $tf(f(u_1, f(u_2, u_3))))$ be $\langle w_1, \ldots, w_n \rangle$ with $n \geq 3$. Therefore, as u_1, u_2 and u_3 are irreducible, $tf(s\sigma) = \langle w_i, \ldots, w_j \rangle$ for some p and q with $1 \leq i < j \leq n$ and $tf(s'\sigma) = \langle w_k, \ldots, w_r \rangle$ for some k and r with $1 \leq k < r \leq n$.

Now there are several cases to be considered, depending on the order relation between i, j, k and r.

If $i = k \leq r < j$ then $s\sigma =_A f(s'\sigma, v_5)$ with $tf(f(s'\sigma, v_5)) = \langle w_i, \ldots, w_j \rangle$, and therefore the following inference by A-strict superposition right 1

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

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 $\theta = \sigma \cup \{x \mapsto v_5\}$.

$D_1\theta$ is variable irreducible wrt. R_S since $D'\theta$ and $D\theta$ are variable irreducible wrt. R_S (by lemma 15) 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 A-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 x, then, since v_5 must be a proper subterm of u for some $u =_A s\sigma$, v_5 is irreducible wrt. R_S (as the rule $s\sigma \Rightarrow t\sigma$ has 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, $R_S \cup A^{\prec D\theta} \cup Ir_{R_S}(S)^{\prec D\theta} \models D_1\theta$ (as $\to_{R_S} \subseteq \succ$ by construction), so from the minimality of C (note that $C \succ D\theta$) we have $I_S \models D_1\theta$. We know $I_S \not\models \Gamma'\sigma, \Gamma\sigma \to \Delta'\sigma, \Delta\sigma$, so it must be the case that $I_S \models f(t', x)\theta \simeq t\theta$, i.e. $I_S \models f(t'\sigma, v_5) \simeq t\sigma$, but since

$s_1 =_{A^{\prec C}} f(w_1, f(\ldots, f(w_{j-1}, f(\quad t\sigma \quad, f(w_{r+1}, f(\ldots, f(w_{n-1}, w_n))))) \ldots))$ and
$s_2 =_{A^{\prec C}} f(w_1, f(\ldots, f(w_{j-1}, f(f(t'\sigma, v_5), f(w_{r+1}, f(\ldots, f(w_{n-1}, w_n))))) \ldots))$,
it contradicts $I_S \not\models C$.

Similarly if $i < k \leq j = r$ another A-strict superposition inference is needed (in this case with $f(x, s')$instead of $f(s', x)$) and when $i < k \leq j < r$ we apply an A-top superposition rule. In all other cases a contradiction is directly obtained.

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 a maximal equation in its succedent.
2. C has a maximal equation $s\sigma \simeq t\sigma$ in its antecedent, with $s\sigma =_A t\sigma$.
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 $s\sigma \neq_A t\sigma$, since if C consists only of the equation $s\sigma \simeq t\sigma$ then $s\sigma =_A t\sigma$ would follow from $A^{\prec C}$ plus instances of A that are A-equal to C, which are shown true in I_S by the first part of this proof; otherwise in fact $s\sigma =_{A^{\prec C}} t\sigma$, which, since $I_S \models A^{\prec C}$, 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 10 do not hold:

If condition 1.b) does not hold then an inference by factoring can be made and its conclusion contradicts the assumption.

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 $s\sigma =_A u$ 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 A or non-A 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.

Let C' be a non-extended instance $D'\sigma$ of some clause $D' \llbracket T' \rrbracket$ (the case where C' is an extended instance is proved similarly) with D' of the form $\Gamma' \rightarrow \Delta', s' \simeq t'$, where the rule is $u' \Rightarrow t'\sigma$ for some u' with $s'\sigma =_A u' = u|_p$ (we can use the same σ since D and D' do not share variables). Two cases have to be considered:

(a) If $maxpos(u, p) = p$ then $u' =_A s\sigma|_{p'}$ for the corresponding maximal position p' in $s\sigma$ and therefore $s'\sigma =_A s\sigma|_{p'}$. 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 7 does not apply either, since $maxpos(s\sigma, p') = p'$. Then the inference by A-strict superposition right

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

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.

(b) If $p_1 = maxpos(u, p) \neq p$ then $u|_{p_1} =_A s\sigma|_{p'}$ for the corresponding maximal position p' in $s\sigma$ and either (i) $u|_{p_1} =_A f(u', v) =_A f(s'\sigma, v)$ or (ii) $u|_{p_1} =_A f(v, u') =_A f(v, s'\sigma)$ or (iii) $u|_{p_1} =_A f(v, f(u', v')) =_A f(v, f(s'\sigma, v'))$, for some v and v', and therefore either (i) $s\sigma|_{p'} =_A f(s'\sigma, v)$ or (ii) $s\sigma|_{p'} =_A f(v, s'\sigma)$ or (iii) $s\sigma|_{p'} =_A f(v, f(s'\sigma, v'))$. Then $s\sigma|_{p'}$ cannot be below a variable, as before, since again $maxpos(s\sigma, p') = p'$. These three cases correspond to three different A-strict superposition right inferences that can be made, respectively with the extension $f(s', x)$, $f(x, s')$ and $f(x, f(s', x))$, which conclusions contradict the assumption.

2. When C has a maximal equation in its antecedent whose members are A-equal, then an following inference by equality resolution can be made and its conclusion contradicts the assumption.

3. If $D\sigma$ 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 13 $s\sigma$ 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: 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 A-superposition left are considered instead of superposition right. □

Now our aim will be to compute saturated 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 18. Let R be a ground rewrite system.

1. A *theorem proving derivation* is a sequence of sets of constrained clauses $S_0, S_1, \ldots$, such that each S_{i+1} is obtained from S_i by adding a logical consequence of S_i or by removing a clause $C \llbracket T \rrbracket$ that is redundant in $S_i \setminus C \llbracket T \rrbracket$.
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 is *fair* if every inference by I_A with persisting premises is redundant in some S_j.

Lemma 19. *Let $S_0, S_1, \ldots$ be a theorem proving derivation, let π be an inferences and let $C \llbracket T \rrbracket$ a constrained clause. (i) If π is redundant in some S_j then π is redundant in S_∞, and (ii) if $C \llbracket T \rrbracket$ is redundant in some S_j then $C \llbracket T \rrbracket$ is redundant in S_∞, for $j \geq 0$.*

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 20. *If $S_0, S_1, \ldots$ is a fair theorem proving derivation and S_0 is a set containing only clauses with empty contraints, then (i) S_∞ is saturated, and (ii) either the empty clause is in some S_j (hence S_0 is inconsistent) or else $I_{S_\infty} \models S_j \cup A$ for $j \geq 0$ (hence S_0 is consistent) and $I_{S_\infty} \models S_\infty \cup A$.*

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 19, π is also redundant wrt. R_{S_∞} in S_∞.

(ii) by lemma 17 (applied to S_∞ and R_{S_∞}) and (i), we have that either the empty clause is in S_∞ or else $I_{S_\infty} \models Ir_{R_{S_\infty}}(S_\infty) \cup A$. 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 19, we have $R_{S_\infty} \cup Ir_{R_{S_\infty}}(S_\infty) \cup A \models Ir_{R_{S_\infty}}(S_0)$, and, since all clauses in S_0 have empty contraints, $R_{S_\infty} \cup Ir_{R_{S_\infty}}(S_0) \models S_0$. Therefore, as $I_{S_\infty} \models R_{S_\infty}$, it follows that $I_{S_\infty} \models S_0 \cup A$ (which proves the consistency of S_0). Furthermore, as by soundness of the derivation $S_0 \cup A \models S_j$ for $j \geq 0$ and $S_0 \cup A \models S_\infty$, we have $I_{S_\infty} \models S_j \cup A$ for $j \geq 0$ and $I_{S_\infty} \models S_\infty \cup A$. □

Recall that $C \llbracket T \rrbracket$ 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 A-equality constraints are essential for soundness. Therefore we only have to decide the satisfiability of the A-equality part of T to know whether $C \llbracket T \rrbracket$ 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. For instance, we can simply check whether there is some $s \succ t$ in the ordering constraint s.t. $t \succ_A s$, which by stability under subtitutions of $\succ_A$ implies the unsatisfiability of the constraint. On the other hand we can look for loops or contradictions with the monotonicity or subterm properties.

Note that, since the word problem modulo associativity is undecidable even in ground equational theories, there may be ground sets of clauses for which there is no finite saturated system, and for which derivations indeed do not terminate.

Finally, let us remark that this result also holds when certain initial constraints are allowed. This problem is studied in [15].

5.1 Redundancy tests

First of all note that in this framework a simplification step, replacing a clause C by its simplified clause C', can be performed with two steps in the derivation: the first one adding C' (which is a consequence) and the second one deleting C (which becomes redundant).

Therefore, we will concentrate in giving sufficient conditions to ensure the redundancy of clauses and inferences.

Lemma 21. *Let S be a set of constrained clauses. A clause $C \llbracket T \rrbracket$ is redundant in S if for every ground instance $C\sigma$ of it, there are ground instances $e_1 \ldots e_r$ of $A^{\preceq C\sigma}$ and ground instances $D_i\sigma_i$ in $S^{\preceq C\sigma}$ of clauses $D_i \llbracket T_i \rrbracket$ with $\{e_1 \ldots e_r, D_1\sigma_1, \ldots, D_m\sigma_m\} \models C\sigma$ and T_i is true for all $i = 1 \ldots m$.*

This result can be slightly strengthened (see [15] or [12] for the AC-case), but for simplicity reasons we present it here like this. Also note that it is straightforward to adapt this lemma for redundancy of inferences.

With respect to practice, first of all, note that for the "classical" saturation methods without constraint inheritance, the previous lemma shows that the usual redundancy notions are correct, since the condition on the constraints is always true, i.e. a clause C is redundant in S if $S^{\preceq C\sigma} \cup A^{\preceq C\sigma} \models C\sigma$ 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. On the other hand, our concept of redundant clause allows to include subsumption without the need of combining the underlying orderings with special subsumption orderings.

In the constraint case, if the condition on the constraints of the previous lemma fails for some constraint T_i of a clause D_i, and we still want to carry out the redundancy proof, then sometimes we can make T_i to be true (ensuring soundness) by propagating the information of the literals in the equality constraint part. Note that if we have started with an initial set of clauses without ordering constraints we can always delete the ordering constraint part without loosing soundness. 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.

6 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 $I_S \models S \cup A$, 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* A, denoted $\rightarrow_{E/\mathrm{A}}$, and for *extended* A*-rewriting* denoted $\rightarrow_{E\backslash\mathrm{A}}$. Let $l\sigma \simeq r\sigma$ be an instance of an equation of E with $l\sigma \succ r\sigma$. Then $s \rightarrow_{E/\mathrm{A}} u[r\sigma]$ if $s =_A u[l\sigma]$. Furthermore, extended A-rewriting is defined as $s[u] \rightarrow_{E\backslash\mathrm{A}} s[f(x, f(r, x'))\sigma]$ if $u =_A f(x, f(l, x'))\sigma$ (with x and x' possibly empty and $f = top(l)$).

Theorem 22. *Let E be a saturated set of constrained equations. For all terms s and t*
$E \cup A \models s \simeq t$ *iff* $s \rightarrow^*_{E/\mathrm{A}} \circ \leftrightarrow^*_{\mathrm{A}} \circ \leftarrow^*_{E/\mathrm{A}} t$ *iff* $s \rightarrow^*_{E\backslash\mathrm{A}} \circ \leftrightarrow^*_{\mathrm{A}} \circ \leftarrow^*_{E\backslash\mathrm{A}} t$.

Let us remark that if r has no "extra variables" (variables that are not in l), then A-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 $(E \cup A)$-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 A-unifiers may even be unnecessary, but this has to be studied in detail. Note that this is crucial here since if the A-unification problem has an infinite minimal complete set of A-unifiers then we may not be able to find the needed substitution.

A similar decision result (by refutation or by conditional rewriting) holds for conditional equations or Horn clauses.

Due to this relation with Knuth-Bendix completion modulo A, the application of these techniques to completion of string rewriting systems or, in particular, to Thue systems should be studied. Note that the presence of equality constraints could be important to obtain finite complete systems, since they can capture some divergences.

References

1. Leo Bachmair. *Canonical equational proofs.* Birkhäuser, Boston, Mass., 1991.
2. Leo Bachmair and Nachum Dershowitz. 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, 1987. Springer-Verlag.
3. Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):1–31, 1994.
4. Nachum Dershowitz and Jean-Pierre Jouannaud. 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, 1990.
5. G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, October 1980.
6. Jean-Pierre Jouannaud. 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, 1983.
7. 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.
8. D. S. Lankford and A. M. Ballantyne. 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, August 1977.
9. G.S. Makanin. The problem of solvability of equations in a free semigroup. *Math. USSR Sbornik*, 32(2):129–198, 1977.

10. 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.
11. 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.
12. Robert Nieuwenhuis and Albert Rubio. AC-Superposition with constraints: No AC-unifiers needed. In Allan Bundy, editor, *12th International Conference on Automated Deduction*, LNAI, Nancy, France, June 1994. Springer-Verlag.
13. Robert Nieuwenhuis and Albert Rubio. Theorem Proving with Ordering and Equality Constrained Clauses. *J. of Symbolic Computation*, 19(4):321–351, April 1995.
14. G.E. Peterson and M.E. Stickel. Complete sets of reductions for some equational theories. *Journal Assoc. Comput. Mach.*, 28(2):233–264, 1981.
15. Albert Rubio. Automated deduction with ordering and equality constrained clauses. PhD. Thesis, Technical University of Catalonia, Barcelona, Spain, 1994.
16. Albert Rubio and Robert Nieuwenhuis. A total AC-compatible ordering based on RPO. *Theoretical Computer Science*, 142(2):209–227, May 15, 1995.
17. Laurent Vigneron. Associative Commutative Deduction with constraints. In Allan Bundy, editor, *12th International Conference on Automated Deduction*, LNAI, Nancy, France, June 1994. Springer-Verlag.

This article was processed using the LaTeX macro package with LLNCS style