Redundancy Notions for Paramodulation with Non-Monotonic Orderings

Miquel Bofill[1] and Albert Rubio[2] *

[1] Universitat de Girona, Dept. IMA, Girona, Spain
mbofill@ima.udg.es

[2] Universitat Politècnica de Catalunya, Dept. LSI, Barcelona, Spain
rubio@lsi.upc.es

Abstract. Recently, ordered paramodulation and Knuth-Bendix completion were shown to remain complete when using non-monotonic orderings. However, these results only implied the compatibility with too weak redundancy notions and, in particular, demodulation could not be applied at all.

In this paper, we present a complete ordered paramodulation calculus compatible with powerful redundancy notions including demodulation, which strictly improves previous results.

Our results can be applied as well to obtain a Knuth-Bendix completion procedure compatible with simplification techniques, which can be used for finding, whenever it exists, a convergent TRS for a given set of equations and a (possibly non-totalizable) reduction ordering.

1 Introduction

Knuth-Bendix-like completion techniques and their extensions to ordered paramodulation for first-order clauses are among the most successful methods for automated deduction with equality [BG98,NR01]. For many years, all known completeness results for Knuth-Bendix completion and ordered paramodulation required the term ordering $\succ$ to be well-founded, monotonic and total (or extendible to a total ordering) on ground terms [HR91,BDH86,BD94,BG94]. In [BGNR99,BGNR03], the monotonicity requirement was dropped and well-foundedness and the subterm property were shown to be sufficient for ensuring refutation completeness of ordered paramodulation (note that any such ordering can be totalized without losing these two properties). And recently, in [BR02], it was shown that well-foundedness of the ordering suffices for completeness of ordered paramodulation for Horn clauses, i.e., the subterm property can be dropped as well. Apart from its theoretical value, these results have several potential applications in contexts where the usual requirements are too strong.

* Both authors partially supported by the Spanish CICYT project MAVERISH ref. TIC2001-2476-C03-01. M. Bofill is also supported by the Spanish CICYT project CADVIAL ref. TIC2001-2392-C03-01 and A. Rubio is also supported by the Spanish DURSI group 2001SGR 00254.

For example, in deduction modulo built-in equational theories E, where E-*compatibility* of the ordering (i.e. $s =_E s' \succ t' =_E t$ implies $s \succ t$) is needed, finding E-*compatible* orderings fulfilling the required properties is extremely complex or even impossible. For instance, when E contains an idempotency axiom $f(x, x) \simeq x$, no total E-compatible reduction ordering exists: if $s \succ t$, by monotonicity one should have $f(s, s) \succ f(s, t)$, which by E-compatibility implies $s \succ f(s, t)$, and hence non-well-foundedness. Therefore, the techniques for dropping ordering requirements, among other applications, open the door to deduction modulo many more classes of equational theories.

However, ordered strategies are useful in practice only if compatibility with redundancy is shown. Simplification of formulae and elimination of redundant formulae are essential components of automated theorem provers. In fact, in most successful automated theorem provers, simplification is the primary mode of computation, whereas prolific deduction rules are used only sparingly.

Here, we present a calculus which strictly improves the one in [BGNR03] (due to lack of space, we restrict ourselves to the equational case). On the one hand, regarding the amount of inferences needed to be performed, our inference system is essentially the same as the one in [BGNR03], but, on the other hand, our calculus is compatible with powerful redundancy notions which include simplification by rewriting (i.e. demodulation). As a side effect, as in [BGNR03], we can apply our results to obtain a Knuth-Bendix completion procedure, but in this case compatible with simplification techniques. This procedure can be used for finding, whenever it exists, a convergent TRS for a given set of equations and a (possibly non-totalizable) reduction ordering.

In our calculus it is assumed that there is a known reduction ordering $\succ_r$, which is included in the (possibly non-monotonic) ordering $\succ$ w.r.t. which equations are oriented. Then we show that redundancy notions w.r.t. the reduction ordering $\succ_r$ can be applied while keeping refutation completeness. Roughly, a formula is redundant if it is a logical consequence of smaller (w.r.t. a well-founded ordering including $\succ_r$) formulae. Then demodulation w.r.t. $\succ_r$ fits in this abstract notion of redundancy since, in particular, a formula which has been simplified by rewriting follows from its smaller w.r.t. $\succ_r$ (due to monotonicity of $\succ_r$) "simplified version" and the (also smaller) equations used to simplify it. As a particular case, if all equations involved in the saturation process[3] turn out to be orientable with $\succ_r$, then demodulation can be fully applied. The following well-known simple challenge example illustrates this situation:

Example 1. Under our inference system,

$$f(f(x)) \simeq f(g(f(x)))$$

is saturated if $f(f(x)) \succ_r f(g(f(x)))$, and the ordering $\succ$ w.r.t. which ordered paramodulation is applied is an extension of $\succ_r$ (which cannot be totalized without losing monotonicity, since $g(f(x))$ and $f(x)$ should be incomparable

[3] The saturation of a set of equations (or clauses) S amounts to the closure of S under the inference system *up to redundancy.*

in any monotonic extension). The only possible inferences are (i) an ordered paramodulation with $f(f(x)) \simeq f(g(f(x)))$ on itself at the underlined position of $f(\underline{f(x)})$, which has $f(f(g(f(x)))) \simeq f(g(f(f(x))))$ as conclusion, and (ii) an ordered paramodulation with $f(f(x)) \simeq f(g(f(x)))$ on itself at the underlined position of $f(g(\underline{f(x)}))$, which has $f(f(f(x))) \simeq f(g(f(g(f(x)))))$ as conclusion. In either case, the conclusion can be simplified into a tautology by rewriting it with the equation $f(f(x)) \simeq f(g(f(x)))$ (oriented with the reduction ordering). Therefore, the system is saturated. It is worth noting that, without considering demodulation, the closure of $f(f(x)) \simeq f(g(f(x)))$ under ordered paramodulation w.r.t. $\succ$ diverges. $\square$

We must point out that, although demodulation w.r.t. $\succ_r$ is allowed in our inference system, not every equation can be demodulated: during our saturation process, some subterms of equations become "blocked" for demodulation. Roughly, we block for demodulation those subterms that occur as the small side (w.r.t. $\succ$) of an equation that cannot be handled with $\succ_r$. Later inferences will inherit these blockings. Note that, if paramodulation is applied w.r.t. a non-monotonic ordering $\succ$ and we use full demodulation w.r.t. $\succ_r$, we can lose completeness as shown in Example 4 of Sect. 5.

The reason for adding these blockings comes from our completeness proof, which is based on the model generation technique of [BG94] and its variant used in [BGNR03]. In the latter, in contrast to the former, the ordering used for orienting the equations in the inference system and the ordering used for induction in the completeness proof do not coincide. Concretely, equations are oriented w.r.t. a non-monotonic ordering $\succ$, whereas completeness is proved by induction w.r.t. $\xrightarrow{+}_R$, where R is the limit ground rewrite system that defines the model. This is the key for the completeness proof of ordered paramodulation with non-monotonic orderings given in [BGNR03]. Since completeness is proved by induction w.r.t. $\xrightarrow{+}_R$, the redundancy notions should be defined w.r.t. $\xrightarrow{+}_R$ as well, instead of w.r.t. $\succ$, which is what would be desirable in principle. Unfortunately, $\xrightarrow{+}_R$ is unknown during the saturation process, and it is not clear how it can be approximated sufficiently. This is why in [BGNR03] it was left open to what extent demodulation could be applied (although some practical redundancy notions like tautology deletion and subsumption were shown to preserve completeness).

Here our aim is to add for instance demodulation w.r.t. the reduction ordering $\succ_r$, which is included in the (possibly) non-monotonic extension $\succ$. The idea for doing that is, roughly, to find a well-founded ordering $\succ_i$, which combines as much as possible $\succ_r$ with $\xrightarrow{+}_R$, and then prove completeness by induction w.r.t. $\succ_i$. The problem is that, although $\succ_r$ is a well-founded and monotonic ordering and R is a terminating TRS whose rules are included in a well-founded extension $\succ$ of $\succ_r$, the relation $\rightarrow_R \cup \succ_r$ is not well-founded in general and, in fact, $\xrightarrow{+}_R$ can even contradict $\succ_r$. Take e.g. $f(b) \succ_r f(a)$ and a well-founded extension $\succ$ of $\succ_r$ such that $f(b) \succ f(a) \succ a \succ b$. Then possibly $a \rightarrow b \in R$ (since the rules in R are oriented w.r.t. $\succ$) and hence the infinite sequence $f(a) \rightarrow_R f(b) \succ_r f(a) \cdots$ can be built.

The idea to circumvent this non-well-foundedness problem is to *block* the "bad" right-hand sides, i.e., the ones that can be introduced by a rewriting step with R which is not included in $\succ_r$. Then we can combine $\succ_r$ with $\rightarrow_R$ if the comparisons with $\succ_r$ cannot take into account the blocked terms. For instance, in the previous example, b should be blocked, and then $f(b)$ with b blocked is no longer greater than $f(a)$.

A convenient way to represent terms with blocked positions is by means of superindexed subterms, also called *marked terms*. For example $f(b^x)$, where x is a variable, denotes the term $f(b)$ where the subterm b is blocked. Then, although for performing inferences $f(b^x)$ still corresponds to the term $f(b)$, for the redundancy notions it will be seen as $f(x)$ with a blocked term b in x. In this context $f(b^x)$ cannot be simplified with $f(b) \rightarrow f(a)$, but it could still be simplified with $f(x) \rightarrow g(x)$ into $g(b^x)$, if $f(x) \succ_r g(x)$. Marked terms resemble the term closures of [BGLS95]. However, their semantics is fairly different: here the blockings only have effect on the redundancy notions, i.e., the ordered paramodulation inferences are applied at both blocked and unblocked positions.

Detection of terms that have to be marked and introduction of marks in the equations will be part of our inference system. The terms to be marked come from the equations that cannot be oriented with $\succ_r$ (since we do not know in advance which equations will be part of R). These terms are kept in a set, which is used for introducing marks in the equations. Hence, if there are many equations that cannot be handled by the reduction ordering then many terms will become blocked and we will have almost no redundancy, but the same amount of inferences by paramodulation as in [BGNR03]. On the other hand, if many equations can be handled by the reduction ordering, then few terms will be blocked and we will have as much redundancy as in the paramodulation calculi for total reduction orderings.

The paper is structured as follows. Preliminaries are presented in Sect. 2. Marked terms and orderings on marked terms are defined in Sect. 3. In Sect. 4, our calculus, including the notions of redundancy for inferences, is presented for deduction with sets of equations. In Sect. 5 some practical notions of redundancy are presented. In Sect. 6 we briefly outline how, from our results, a Knuth-Bendix completion procedure for finding convergent TRSs can be obtained. Finally in Sect. 7 we give some conclusions. Due to lack of space, we have omitted many details and all proofs (see [BR04] for the long version of this paper).

2 Preliminaries

We use the standard definitions of [NR01]. $T(\mathcal{F}, \mathcal{X})$ ($T(\mathcal{F})$) is the set of (ground) terms over a set of symbols $\mathcal{F}$ and a denumerable set of variables $\mathcal{X}$ (over $\mathcal{F}$). The subterm of t at *position* p is denoted by $t|_p$, the result of replacing $t|_p$ by s in t is denoted by $t[s]_p$, and syntactic equality of terms is denoted by $\equiv$. A *substitution* is a partial mapping from variables to terms. The application of a substitution σ to a term t is denoted by $t\sigma$. An *equation* is a multiset of terms $\{s, t\}$, denoted $s \simeq t$ or, equivalently, $t \simeq s$.

If $\rightarrow$ is a binary relation, then $\leftarrow$ is its inverse, $\leftrightarrow$ is its symmetric closure, $\xrightarrow{+}$ is its transitive closure and $\xrightarrow{*}$ is its reflexive-transitive closure. If $s \xrightarrow{*} t$ and there is no t' such that $t \rightarrow t'$ then t is called *irreducible* and a *normal form* of s (w.r.t. $\rightarrow$). The relation $\rightarrow$ is *well-founded* or *terminating* if there exists no infinite sequence $s_1 \rightarrow s_2 \rightarrow \dots$ and it is *confluent* or *Church-Rosser* if the relation $\xleftarrow{*} \circ \xrightarrow{*}$ is contained in $\xrightarrow{*} \circ \xleftarrow{*}$. A relation $\rightarrow$ on terms is *monotonic* if $s \rightarrow t$ implies $u[s]_p \rightarrow u[t]_p$ for all terms s, t and u and positions p. An *equivalence relation* is a reflexive, symmetric and transitive binary relation. A *congruence* is a monotonic equivalence relation.

A *rewrite rule* is an ordered pair of terms (s, t), written $s \rightarrow t$, and a set of rewrite rules R is a *term rewrite system* (TRS). The *rewrite relation* with R on $T(\mathcal{F}, \mathcal{X})$, denoted by $\rightarrow_R$, is the smallest monotonic relation such that for all rules $l \rightarrow r$ in R, and substitutions σ, $l\sigma \rightarrow_R r\sigma$. If $s \rightarrow_R t$ then we say that *s rewrites into t* with R. R is called terminating, confluent, etc. if $\rightarrow_R$ is. It is called *convergent* if it is confluent and terminating. The congruence $\xleftrightarrow{*}_R$ defines an *equality* Herbrand interpretation where $\simeq$ is interpreted by $s \simeq t$ iff $s \xleftrightarrow{*}_R t$. Such an interpretation will be denoted by R^*.

A (strict partial) *ordering* $\succ$ on $T(\mathcal{F}, \mathcal{X})$ is an irreflexive and transitive binary relation. It is *stable under substitutions* if $s \succ t$ implies $s\sigma \succ t\sigma$ for all substitutions σ. Monotonic orderings that are stable under substitutions are called *rewrite orderings.* A *reduction ordering* is a well-founded rewrite ordering.

A rewrite system terminates if, and only if, its rules are contained in a reduction ordering. In fact, if R is a terminating TRS, then $\xrightarrow{+}_R$ is a reduction ordering. However in restricted cases checking the rules with a well-founded ordering is enough:

Lemma 1. *[BGNR03] Let $\succ$ be a well-founded ordering, and let R be a ground TRS such that, for all $l \rightarrow r$ in R, $l \succ r$ and r is irreducible by R at non-topmost positions. Then R is terminating.*

The multiset extension of an equivalence relation $\sim$ is denoted by $\sim^{mul}$. The multiset extension of $\succ$ with respect to $\sim$ is defined as the smallest ordering $\succ^{mul}$ on multisets of elements such that $M \cup \{s\} \succ^{mul} N \cup \{t_1, \dots, t_n\}$ if $M \sim^{mul} N$ and $s \succ t_i$ for all i in $1 \dots n$. If $\succ^{mul}$ is used without explicitly indicating which is the equivalence relation $\sim$ then $\sim$ is assumed to be the syntactic equality relation. If $\succ$ is well-founded on a set S, so is $\succ^{mul}$ on finite multisets over S.

There are some orderings that play a central role in our results (in particular, in the redundancy notions). The *subsumption* relation, denoted by $\geq$, is defined as $s \geq t$ if s is an instance of t, i.e., $s \equiv t\sigma$ for some σ. The equivalence relation associated with $\geq$ is denoted by $\doteq$. Note that if $s \doteq t$ then t is a variable renamed version (or a *variant*) of s. The *encompassment* relation, denoted by $\trianglerighteq$, is defined as $s \trianglerighteq t$ if a subterm of s is an instance of t, i.e., $s \trianglerighteq t\sigma$ for some σ; therefore, encompassment is the composition of the subterm and the subsumption relations. We have that, if $\succ_r$ is a reduction ordering, then $\triangleright \cup \succ_r$ is well-founded.

It is said that an ordering fulfills the *subterm property* if $\succ \supseteq \triangleright$, where $\triangleright$ denotes the strict subterm relation. A *west ordering* [BGNR03] is a well-founded ordering on $T(\mathcal{F})$ that fulfills the subterm property and that is total on $T(\mathcal{F})$ (it is called *west* after well-founded, subterm and total). Every well-founded ordering can be totalized [Wec91], and hence every well-founded ordering satisfying the subterm property can be extended to a west ordering. We also have that every reduction ordering can be extended to a west ordering [BGNR03].

3 Marked Terms

Marked terms resemble the closures of [BGLS95]. A *term closure*, following the definition of [BGLS95], is a pair $s \cdot \gamma$ consisting of a term s (the *skeleton*) and a substitution γ from variables to terms. For example, $f(x, g(y)) \cdot \{x \mapsto a, y \mapsto h(z)\}$ is a closure with skeleton $f(x, g(y))$ and substitution $\{x \mapsto a, y \mapsto h(z)\}$. We generalize closures in that the substitution can map a variable not only to a term but also to a closure. For example, $f(x, g(y)) \cdot \{x \mapsto a, y \mapsto h(z) \cdot \{z \mapsto b\}\}$ is a generalized term closure, called a *marked term*. Moreover, closures are used in [BGLS95] to express that certain positions are blocked for inferences, whereas in our case they are blocked only for redundancy purposes. In order to ease the reading, sometimes we will denote marked terms by superindexing the "marked" subterms with variables. For example,

$$f(x, g(y)) \cdot \{x \mapsto a, y \mapsto h(z) \cdot \{z \mapsto b\}\} \text{ will be written as } f(a^x, g(h(b^z)^y)).$$

An equation between marked terms is called a *marked equation*. Given a substitution γ of the form $\{x_1 \mapsto t_1 \cdot \gamma_1, \ldots, x_n \mapsto t_n \cdot \gamma_n\}$, the domain of γ, denoted by $Dom(\gamma)$, is defined as the set of variables $\{x_1, \ldots, x_n\}$, and the range of γ, denoted by $Ran(\gamma)$, is defined as the set of marked terms $\{t_1 \cdot \gamma_1, \ldots, t_n \cdot \gamma_n\}$. The variables occurring in $Dom(\gamma)$ are called *marking variables*. We will restrict marked terms to those which fulfill that marking variables occur at most once in the skeleton. For example $f(x, y) \cdot \{x \mapsto a, y \mapsto a\}$ is a valid marked term, but $f(z, z) \cdot \{z \mapsto a\}$ is not. Therefore, although we generalize closures in one sense, in another sense we restrict them.[4]

The congruence $\equiv$ on marked terms is defined as the monotonic, reflexive, symmetric and transitive closure including (i) $t^{xy} \equiv t^x$, and (ii) $s[t^x]_p \equiv s[t^y]_p$ if neither x nor y occur in $s[t]_p$.[5] Note that if $s \cdot \gamma \equiv t \cdot \delta$, then $s \doteq t$, i.e., s is a variant of t.

Let $s \cdot \{x_1 \mapsto t_1 \cdot \gamma_1, \ldots, x_n \mapsto t_n \cdot \gamma_n\}$ be a marked term and σ be a substitution of the form $\{y_1 \mapsto u_1 \cdot \delta_1, \ldots, y_m \mapsto u_m \cdot \delta_m\}$, such that the variables in the domains of all substitutions are distinct. Then $(s \cdot \gamma)\sigma$ is defined as

$$s\{y_1 \mapsto u_1, \ldots, y_m \mapsto u_m\} \cdot (\{x_1 \mapsto (t_1 \cdot \gamma_1)\sigma, \ldots, x_n \mapsto (t_n \cdot \gamma_n)\sigma\} \cup \delta_1 \cup \cdots \cup \delta_m),$$

[4] Both the generalization and the restriction are necessary for our orderings on marked terms being well-founded.

[5] There is no condition that prevents terms like t^{xy} from appearing. For example, a marked variable z in e.g. z^y may be instantiated with a marked term t^x, giving t^{xy}. However, we will assume that these variable-variable bindings are instantiated whenever they arise, i.e. that terms are kept in its simpler form (t^x in the example).

where the marking variables of the resulting skeleton are renamed if necessary to keep linearity. For example, $f(x, g(x)^z)\{x \mapsto a^y\} = f(a^y, g(a^y)^z)$, and $f(x, x)\{x \mapsto a^y\} = f(a^y, a^z)$.

By $\mathit{Forget}(s \cdot \gamma)$ we denote the term in $T(\mathcal{F}, \mathcal{X})$ obtained from $s \cdot \gamma$ by repeatedly applying all the substitutions. For example, $\mathit{Forget}(f(a^y, g(a^y)^z))$ is $f(a, g(a))$. A marked term $s \cdot \gamma$ is said to be ground if $\mathit{Forget}(s \cdot \gamma)$ is a ground term. If $(s \cdot \gamma)\sigma$ is a ground marked term, then σ is called a *ground substitution* for $s \cdot \gamma$, and $(s \cdot \gamma)\sigma$ is called a *ground instance* of $s \cdot \gamma$. This notation is extended to sets of terms and equations in the usual way.

The usual concepts of (and notation for) position, subterm and subterm replacement can be easily extended for marked terms. For example, $f(x, g(x)^z)|_2 = g(x)^z$, and $f(x, g(x)^z)[a^y]_{2\cdot 1} = f(x, g(a^y)^z)$.

From these definitions, the notion of rewriting for terms can be extended to marked terms in the natural way, as well as the concept of unifier: a unifier of two marked terms $s \cdot \gamma$ and $t \cdot \delta$ is a substitution σ such that $(s \cdot \gamma)\sigma \equiv (t \cdot \delta)\sigma$. There exists a unique (up to renaming of variables) most general unifier between marked terms, which can be obtained by straightforwardly adapting any known unification algorithm for terms.

3.1 Orderings on Marked Terms

Let $\succ_r$ be a reduction ordering, $\succ$ be a west ordering including $\succ_r$, and R be a terminating ground TRS. In this section we define the two main orderings on marked terms, namely $\succ_{red}$, which will be used in the redundancy notions and will be closely related to $\succ_r$, and $\succ_i$, which will be used for induction in the completeness proof and will combine as much as possible $\succ_{red}$ with $\xrightarrow{+}_R$. For the reasons explained in the introduction, it is crucial for the completeness proof that $\succ_{red} \subseteq \succ_i$.

We first define the marking ordering $\succ_m$. It only applies to terms that just differ in the markings and it is similar to the subsumption ordering for terms, in the sense that a term is smaller if it has higher marks.

Definition 1. *Let $s \cdot \gamma$ and $t \cdot \delta$ be two marked terms. Then $s \cdot \gamma \succ_m t \cdot \delta$ iff $\mathit{Forget}(s \cdot \gamma) \equiv \mathit{Forget}(t \cdot \delta)$, $s \cdot \gamma \not\equiv t \cdot \delta$ and (i) $s \gtrdot t$ or (ii) $s \doteq t$ and $\mathit{Ran}(\gamma) \succ_m^{mul} \mathit{Ran}(\delta)$.*

Since $\succ_{red}$ will be used in the redundancy notions, all its ingredients are known while performing paramodulation steps. Given two marked terms, $\succ_{red}$ roughly compares the skeletons w.r.t. the reduction ordering and the encompassment relation and, if the skeletons are the same and the terms only differ on the markings, then it compares the level of the marks in the substitution part.

Definition 2. *Let $s \cdot \gamma$ and $t \cdot \delta$ be two marked terms. Then $s \cdot \gamma \succ_{red} t \cdot \delta$ iff (i) $s \rhd \cup \succ_r t$ or (ii) $s \doteq t$ and $\mathit{Ran}(\gamma) \succ_m^{mul} \mathit{Ran}(\delta)$.*

Since $\succ_i$ will be used for induction in the completeness proof, it extends $\succ_{red}$ with an additional case which uses as ingredient a terminating rewrite system R that will be known only in the proof.

Definition 3. *Let $s \cdot \gamma$ and $t \cdot \delta$ be two marked terms. Then $s \cdot \gamma \succ_i t \cdot \delta$ iff (i) $s \rhd \cup \succ_r t$ or (ii) $s \doteq t$ and $Ran(\gamma) \succ_m^{mul} Ran(\delta)$ or (iii) $s \doteq t$ and $Forget(Ran(\gamma)) \xrightarrow{+}_R Forget(Ran(\delta))$.*

In what follows, we will consider $\succ_i$ and $\succ_{red}$ as the transitive closures of the orderings defined above.

Lemma 2. *$\succ_i$ is well-founded.*

Example 2. Here there are some examples of applications of $\succ_i$.

1. If $f(x) \succ_r g(x)$ for all x, then $f(a) \succ_i f(a^x) \succ_i g(a^x) \succ_i a^x$ by case (i) of the definition of $\succ_i$.
2. We have $h(g(f(a^x, a))^y) \succ_i h(g(f(a^x, a^z))^y) \succ_i h(g(f(a, a)^x)^y)$ by case (ii) of the definition of $\succ_i$.

4 Paramodulation with Equations

In this section we present an ordered paramodulation calculus for sets of equations. The calculus will work on both a set of marked equations E and a set of terms T, which contains the set of terms that are going to be marked.

In the following, let $\succ_r$ be a given reduction ordering and $\succ$ be a given west ordering including $\succ_r$. We assume that the non-marking variables of all equations are disjoint (if necessary, the variables can be renamed). On the other hand, to ease the reading, the marked terms of each equation are considered to have the same substitution (if necessary, the substitution can be extended and the marking variables of one of the two skeletons can be renamed).

Definition 4. *Let E be a set of marked equations and T be a set of terms, which satisfy the following condition: for every marked subterm $(s \cdot \gamma)^x$ occurring in E, we have that $Forget(s \cdot \gamma)$ is an instance of a term in T. The inference system $\mathcal{I}$ consists of the following two inference rules and a mark detection rule:*

Paramodulation*:*

$$\frac{l \cdot \delta \simeq r \cdot \delta \qquad s \cdot \gamma \simeq t \cdot \gamma}{(s \cdot \gamma[r \cdot \delta]_p \simeq t \cdot \gamma)\sigma} \qquad \text{if}$$

1. $\sigma = mgu(l \cdot \delta, s \cdot \gamma|_p)$, the most general unifier of $l \cdot \delta$ and $s \cdot \gamma|_p$,
2. $Forget(s \cdot \gamma|_p)$ is not a variable,
3. $l \succ_r r$ or $r \cdot \delta$ is marked on top, and
4. for some ground substitution θ, we have $Forget((l \cdot \delta)\sigma\theta) \succ Forget((r \cdot \delta)\sigma\theta)$ and, if $p = \lambda$, then we also have $Forget((s \cdot \gamma)\sigma\theta) \succ Forget((t \cdot \gamma)\sigma\theta)$.

Marking*:*

$$\frac{s \cdot \gamma[u \cdot \delta]_p \simeq t \cdot \gamma}{(s \cdot \gamma[(u \cdot \delta)^x]_p \simeq t \cdot \gamma)\sigma} \qquad \text{if}$$

1. $\sigma = mgu(Forget(u \cdot \delta), u')$ *for some term* u' *in* T*,*
2. u *is not a variable, and*
3. x *is a fresh variable.*

Mark detection*:*

$$T \Longrightarrow T \cup \{Forget(r \cdot \delta)\} \qquad \textit{if}$$

1. $l \cdot \delta \simeq r \cdot \delta \in E$*,*
2. $l \not\succ_r r$ *and* $r \cdot \delta$ *is not marked on top, and*
3. *for some ground substitution* θ*, we have that* $Forget((l \cdot \delta)\theta) \succ Forget((r \cdot \delta)\theta)$*.*

Some examples of applications of these rules can be found in Example 4 of Sect. 5.

4.1 Redundancy Notions: a Static View

Here we define some abstract redundancy notions for inferences, like the ones in [BG94] and [NR01]. Some practical redundancy notions fitting in these abstract notions are discussed afterwards, in Sect. 5.

Given a set of terms M, by $gnd_M(E)$ we denote the set of ground instances $e\theta$ of equations e in E such that θ is a substitution with marks in M. Given a well-founded ordering $>$ on marked equations, by $E^{<e}$ we denote the set of all d in E such that $e > d$. Now we define redundancy of inferences and mark detection rules. Some explanations are given below.

Definition 5. *Let* π *be an inference by* $\mathcal{I}$ *with premises* $e_1, \ldots e_n$ *and conclusion* d*. Then a* ground instance $\pi\theta$ of the inference π *is an inference by* $\mathcal{I}$ *with premises* $e_1\theta, \ldots e_n\theta$ *and conclusion* $d\theta$ *for some ground substitution* θ*.*

Let G *be a set of ground marked equations. A ground inference by* $\mathcal{I}$ *with premises* $e_1, \ldots e_n$ *and conclusion* d *is* redundant in G *if*

$$Forget(G^{\prec_{red}^{mul} e_n}) \cup Forget(G^{\preceq_{red}^{mul} d}) \models Forget(d).$$

Let E *be a set of marked equations. An inference by* $\mathcal{I}$ *with premises* $e_1, \ldots e_n$ *and conclusion* d *is* redundant in E *if, for all sets of terms* M *and ground substitutions* θ *with marks in* M*,* $\pi\theta$ *is redundant in* $gnd_M(E)$*.*

Let T *be a set of terms. A mark detection rule* $T \Longrightarrow T \cup \{t\}$ *is redundant in* T *if* t *is an instance of a term in* T*.*

Note that the set of terms M is universally quantified in the previous definition of redundancy for inferences. This is due to the fact that the set of terms T to be marked will change during the computation, and it is crucial to have a notion of redundancy which is preserved along the computation. Moreover, the set $G^{\preceq_{red}^{mul} d}$ may not be included in $G^{\prec_{red}^{mul} e_n}$ since, in general, the conclusion of an inference does not need to be smaller w.r.t. $\succ_{red}^{mul}$ than the rightmost premise.

Redundant inferences and mark detection rules are unnecessary, and therefore we are interested in computing the closure of the set of equations with respect to $\mathcal{I}$ up to redundancy.

Definition 6. *Let E be a set of equations and T be a set of terms. The pair $\langle E, T\rangle$ is* saturated *with respect to $\mathcal{I}$ if every inference by $\mathcal{I}$ is redundant in E and every mark detection rule is redundant in T.*

4.2 Model Generation

Given a set of marked equations E and a set of terms T, here we show how to build an equational model of E defined by means of a set of rules coming from ground instances of E and T. Since the model is defined inductively, we need to treat E and T in a uniform way. To this end, a new constant symbol $\perp$ is introduced and the west ordering is extended with $\perp$ as minimal element.

Let T_{E} denote the set of equations $t \simeq \perp$ for all t in T, and T_{G} denote the set of mark free ground instances of equations in T_{E}. The model will be defined inductively on the equations in T_{G} and $gnd_T(E)$ with respect to $\succ_{mod}$, a total well-founded ordering including: $(s \simeq t) \succ_{mod} (u \simeq v)$ if

1. $Forget(s \simeq t) \succ^{mul} Forget(u \simeq v)$.
2. $Forget(s \simeq t) \equiv^{mul} Forget(u \simeq v)$ and $(s \simeq t) \succ_m^{mul} (u \simeq v)$, which means that $u \simeq v$ has marks at higher positions than $s \simeq t$.

Now we define a rewrite system R_E from $gnd_T(E)$, which requires to define at the same time a rewrite system R_T from T_{G}.

Definition 7. *The rewrite system R_{ET} is the union of R_E and R_T. Let e be a ground (marked) equation in $gnd_T(E) \cup T_{\mathrm{G}}$.*

1. *If e is an instance $(s\cdot\gamma)\theta \simeq (t\cdot\gamma)\theta$ in $gnd_T(E)$ of some equation $s\cdot\gamma \simeq t\cdot\gamma$ in E, then it generates the rule $(s\cdot\gamma)\theta \to (t\cdot\gamma)\theta$ in R_E if*
   (a) $s \succ_r t$ or $t\cdot\gamma$ is marked on top,
   (b) $Forget((s\cdot\gamma)\theta) \succ Forget((t\cdot\gamma)\theta)$,
   (c) $(s\cdot\gamma)\theta$ is irreducible by $R_E^e \cup R_T^e$, and
   (d) $(t\cdot\gamma)\theta$ is irreducible by $R_E^e \cup R_T^e$ at non-topmost positions.
2. *If e is an instance $t\theta \simeq \perp$ in T_{G} of an equation $t \simeq \perp$ in T_{E}, then it generates the rule $u\cdot\delta \to (u\cdot\delta)^x$ in R_T if $u\cdot\delta$ is the normal form of $t\theta$ w.r.t. R_T^e and $u\cdot\delta$ is irreducible by R_E^e,*

where R_E^e and R_T^e denote the set of rules in R_E and R_T respectively, generated by equations d such that $e \succ_{mod} d$.

The following lemma about irreducibility of left and right-hand sides of R_{ET} holds easily by construction.

Lemma 3. *For all rules $l\cdot\delta \to r\cdot\delta$ in R_{ET} we have that*

1. *$r\cdot\delta$ is irreducible by R_{ET} at non-topmost positions, and*
2. *$l\cdot\delta$ is irreducible by $R_{ET} \setminus \{l\cdot\delta \to r\cdot\delta\}$.*

Again by construction of R_{ET}, and by the fact that all marks correspond to instances of terms in T, we obtain the following.

Lemma 4. *Let $s \cdot \gamma$ be a marked subterm occurring, not as right-hand side, in some rule of R_E. Then $s \cdot \gamma$ is marked on top if, and only if, $Forget(s \cdot \gamma)$ is an instance of a term in T.*

Then, from Lemmas 3 and 4, we obtain the same irreducibility result for $Forget(R_E)$ as for R_{ET} since, roughly, forgetting the marks cannot cause reducibility (otherwise, we would already had reducibility in R_{ET}). From this irreducibility result, convergence of $Forget(R_E)$ follows.

Lemma 5. *$Forget(R_E)$ is convergent.*

Finally, we can prove that $Forget(R_E)^*$ is a model of $Forget(E)$.

Theorem 1. *If a pair $\langle E, T\rangle$ is saturated with respect to $\mathcal{I}$, then $Forget(R_E)^* \models Forget(E)$.*

The following is a sketch of the proof. Since we have that $Forget(R_E)$ is convergent[6], the proof is a straightforward adaptation of the one in [BG94,NR01], but where the induction is done w.r.t. $\succ_i^{mul}$, taking $R = Forget(R_E)$ (see Sect. 3.1 for the definition of $\succ_i$). A contradiction is derived from the existence of a minimal w.r.t. $\succ_i^{mul}$ ground instance $e\theta$ in $gnd_T(E)$ of an equation e in E, such that $R^* \not\models Forget(e\theta)$. As usual, we have that $e\theta$ has not generated a rule in R_E. If this is because $e\theta$ is reducible by R_E, then we have an inference by paramodulation and, if $e\theta$ is reducible by R_T, then we have an inference by marking and, in both cases, a contradiction is obtained from minimality of $e\theta$ and saturatedness of $\langle E, T\rangle$.

From $Forget(R_E)^* \models Forget(E)$ and convergence of $Forget(R_E)$, it follows that every (ground) equation following from $Forget(E)$ has a rewrite proof with $Forget(R_E)$. Then, roughly, every procedure that computes a saturated pair $\langle E, T\rangle$ with respect to $\mathcal{I}$ from an initial set of equations E_0, will eventually generate sufficient equations for a rewrite proof of any logical consequence of E_0.

Lemma 6. *Let $\langle E, T\rangle$ be a pair of marked equations and terms that is saturated with respect to $\mathcal{I}$, and $s \simeq t$ be an equation such that $Forget(E) \models s \simeq t$. Then for every ground instance $s\sigma \simeq t\sigma$ of $s \simeq t$ we have that $s\sigma \xrightarrow{*}_R u \xleftarrow{*}_R t\sigma$ for some term u, where R denotes $Forget(R_E)$.*

5 Practical Notions of Redundancy

In this section we provide sufficient conditions for proving redundancy of inferences, which includes simplification by rewriting for those rules included in the reduction ordering $\succ_r$.

Let E be a set of equations. By E_r we denote the set of rules $l \cdot \delta \to r \cdot \delta$ such that the equation $l \cdot \delta \simeq r \cdot \delta$ is in E and $l \succ_r r$. An inference with premises $e_1, \ldots, e_n$ and conclusion d can be shown redundant in E in two steps:

[6] In fact, termination of the rewrite system suffices for the proof of this theorem, that is, confluence is not necessary here.

1. First, d is simplified by rewriting it into d' using marked rules $l \cdot \delta \to r \cdot \delta$ in E_r. In order to ensure that we are only using smaller (w.r.t. $\succ_{red}^{mul}$) instances of equations in E, we have the following sufficient conditions. If d is of the form $s \simeq t$, then (i) we can apply any rewriting step at a non-top position of the skeleton of s (resp. t) obtaining s' (resp. t'), and after that we can apply any rewriting step at any position of the skeleton of s' (resp. t'); (ii) we can apply a rewriting step at top position of s (resp. t) obtaining s' (resp. t') if the instance of the applied rule is smaller (w.r.t. $\succ_{red}^{mul}$) than e_n or smaller than or equal to d, and after that we can apply any rewriting step at any position of the skeleton of s' (resp. t'). Note that in all these cases we can show that for any ground instance of the inference, the ground instances of the applied rules are either smaller (w.r.t. $\succ_{red}^{mul}$) than the instance of e_n or smaller than or equal to the instance of d.
2. Second, we check whether $d_1, \ldots, d_n \models d'$ for some instances of equations in E s.t. $d'(\trianglerighteq_{sk} \cup \succ_m)^{mul} d_i$ for all i in $1 \ldots n$, where $s \cdot \gamma \trianglerighteq_{sk} t \cdot \delta$ if $t \cdot \delta$ occurs in $s \cdot \gamma$ at some position of its skeleton. This includes, among other cases, the case where d' is in E, the case where there is a version of d' with additional marks in E, and the case where d' is a tautology. If this situation does not apply, then we can add d' to E to make the inference redundant.

Example 3. In Example 1 of the introduction we showed how, by an ordered paramodulation with $\underline{f(f(x))} \simeq f(g(f(x)))$ on $f(f(y)) \simeq f(g(\underline{f(y)}))$, we got the equation $f(f(f(x))) \simeq f(g(f(g(f(x)))))$, i.e. $f(f(f(z))) \simeq f(g(f(g(f(z)))))$. In this case, $f(f(f(z)))$ could be rewritten in two steps with $f(f(x)) \to f(g(f(x)))$ into $f(g(f(g(f(z)))))$, and hence the conclusion of the inference could be simplified into a tautology. Observe that the first step is applied at a non-top position, obtaining $f(f(g(f(z))))$, and then the second step can be applied on top obtaining $f(g(f(g(f(z)))))$.

All previous results can be adapted in the usual way for showing *redundancy of clauses* (or equations, in this case), and not only *redundancy of inferences*. Roughly, an equation can be defined as redundant if it follows from smaller equations w.r.t. $\prec_{red}^{mul}$.

In the following example, we show how demodulation w.r.t. $\succ_r$ and redundancy with smaller equations w.r.t. the subterm relation can cause incompleteness if no blockings are introduced at all. This thus shows the need for some control or, in other words, that our blockings are not merely a technicality to prove completeness.

The example has been adapted to the equational case from an example on Horn clauses. In this case, to show that some existentially quantified equation e is a consequence of an equational theory E, a contradiction is derived after adding the negation of e to E. To this end, paramodulation and marking inferences also take place on the negation of e. These paramodulation inferences correspond, at the ground level, to the rewrite steps in the rewrite proof of an instance of e.

Example 4. Let $\succ_r$ be a reduction ordering such that $f(x,x) \succ_r f(a,b)$ and $f(x,x) \succ_r f(b,a)$. Then necessarily $a \not\succ_r b$ and $b \not\succ_r a$. Let E denote the following inconsistent set of equalities and disequalities:

$$\begin{array}{ll}
1) & a \simeq b \\
2) & f(x,x) \simeq f(a,b) \\
3) & f(x,x) \simeq f(b,a) \\
4) & f(x,x) \simeq f(b,b) \\
5) & g(f(a,b)) \simeq g(g(f(b,a))) \\
6) & x \not\simeq g(x)
\end{array}$$

First of all, we will show how a contradiction can be derived under our inference system (which introduces the convenient blockings). Let $\succ$ be a west ordering which is an extension of $\succ_r$ and that includes

$$g(g(f(a,b))) \succ g(f(a,b)) \succ g(g(f(b,a))) \succ g(f(b,a))$$
$$\succ f(a,a) \succ f(b,b) \succ f(a,b) \succ f(b,a) \succ a \succ b.$$

Initially, the set T of terms that are going to be marked is empty. Then, since $a \succ b$ and $a \not\succ_r b$, the *mark detection* rule gives us $T = \{b\}$. After this, an inference by *marking* into 1 gives us

$$7) \quad a \simeq b^{x_0}$$

Now, an inference by *paramodulation* with 7 into 5 is possible (recall that an inference by paramodulation with 1 into 5 is not possible since $a \simeq b$ does not fulfill $a \succ_r b$ and b is not marked on top), giving

$$8) \quad g(f(b^{x_1}, b)) \simeq g(g(f(b,a)))$$

A new inference by *paramodulation* with 7 into 8 gives us

$$9) \quad g(f(b^{x_1}, b)) \simeq g(g(f(b, b^{x_4})))$$

And, with two more inferences by *marking*, we get

$$\begin{array}{ll}
10) & g(f(b^{x_1}, b^{x_2})) \simeq g(g(f(b, b^{x_4}))) \\
11) & g(f(b^{x_1}, b^{x_2})) \simeq g(g(f(b^{x_3}, b^{x_4})))
\end{array}$$

It is worth noting that equation 1 follows from the smaller equation 7, equations 9 and 10 follow from the smaller equation 11, and equation 8 follows from the smaller equations 7 and 11. Hence, using *redundancy of clauses*, the equations 1, 8, 9 and 10 could be deleted.

Now recall that $\succ$ fulfills the subterm property. Therefore we have that $\mathit{Forget}(g(g(f(b^{x_3}, b^{x_4})))) \succ \mathit{Forget}(g(f(b^{x_1}, b^{x_2})))$, and hence an inference by *paramodulation* with 11 into 6 gives us a contradiction,

$$12) \quad g(f(b^{x_1}, b^{x_2})) \not\simeq g(f(b^{x_1}, b^{x_2})).$$

Now let us show that, if blockings were not introduced (in other words, if inferences by *paramodulation* with 1 instead of 7 were allowed), then refutation completeness would be lost. If an inference by *paramodulation* with 1 into 5 was possible, then it would give either $g(f(b,b)) \simeq g(g(f(b,a)))$ or $g(f(a,b)) \simeq g(g(f(b,b)))$, which could be simplified back into 5 respectively with 2 and 3, which are included in the reduction ordering. Moreover, inferences by *paramodulation* with 1 into 2 and 3 give us 4. And inferences by *paramodulation* between 2, 3 and 4 give us already existing equations or equations like $f(a,b) \simeq f(b,a)$, which follow from the smaller (w.r.t. the subterm relation) equation 1. Finally, since $g(f(a,b)) \succ g(g(f(b,a)))$, an inference by *paramodulation* with 5 into 6 gives us

$$7) \quad f(a,b) \not\simeq g(g(f(b,a)))$$

but, since $f(a,b) \simeq g(g(f(b,a)))$ does not follow from the positive equations, further inferences on 7 will not lead to a contradiction. Therefore the set E can be saturated without a contradiction being derived. □

6 Knuth-Bendix Completion

Let E be a set of equations and $\succ_r$ be a (possibly non-totalizable) reduction ordering. Then a *convergent TRS R for E and $\succ_r$* is a convergent TRS, logically equivalent to E, and such that $l \succ_r r$ for all its rules $l \to r$. The problem we deal with here is finding a convergent TRS for the given E and $\succ_r$ whenever it exists, and find it in finite time if it is finite.

The first procedure for this problem, not relying on the enumeration of all equational consequences, was presented in [BGNR99,BGNR03]. From our present results, that method can be improved by making it compatible with powerful redundancy notions, like simplification by rewriting. The proof (given in the long version of this paper [BR04]) is a straightforward adaptation of the one in [BGNR03] since, in the limit, an interreduced convergent TRS is generated and, hence, the same techniques can be applied.

7 Conclusions

We have proposed a complete ordered paramodulation calculus which works with non-monotonic orderings and is compatible with powerful redundancy notions. Our inference system works with a pair of orderings: a west ordering $\succ$ which is used for orienting the equations, and a reduction ordering $\succ_r$, included in $\succ$, which is used in the redundancy notions. Our method is based on adding some marks that block terms for redundancy (but not for inferences). These results extend easily to Knuth-Bendix completion and to Horn clauses, as shown in the long version of this paper [BR04]. We believe that these results extend to general clauses as well, but maybe some more mark detection rules need to be added.

As a final remark, we want to note that a way to obtain, in practice, the orderings $\succ$ and $\succ_r$, is to define the reduction ordering $\succ_r$ with the *monotonic* *semantic path orderings* [BFR00], since then it is directly included, by definition, in a *semantic path ordering* [KL80], which can be used as the west ordering $\succ$.

Acknowledgement. We would like to thank Robert Nieuwenhuis for his helpful comments on this work.

References

[BD94] Leo Bachmair and Nachum Dershowitz. Equational inference, canonical proofs, and proof orderings. *Journal of the ACM*, 41(2):236–276, Feb. 1994.

[BDH86] Leo Bachmair, Nachum Dershowitz, and Jieh Hsiang. Orderings for equational proofs. In *First IEEE Symposium on Logic in Computer Science (LICS)*, Cambridge, Massachusetts, USA, June 1986. IEEE Computer Society Press.

[BFR00] C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. In David McAllester, editor, *Proceedings of the 17th Conference on Automated Deduction (CADE-17)*, volume 1831 of *LNAI*, Pittsburgh, USA, June 2000. Springer-Verlag.

[BG94] Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[BG98] Leo Bachmair and Harald Ganzinger. Equational reasoning in saturation-based theorem proving. In W. Bibel and P. Schmitt, editors, *Automated Deduction: A Basis for Applications*. Kluwer, 1998.

[BGLS95] L. Bachmair, H. Ganzinger, Chr. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[BGNR99] Miquel Bofill, Guillem Godoy, Robert Nieuwenhuis, and Albert Rubio. Paramodulation with non-monotonic orderings. In *14th IEEE Symposium on Logic in Computer Science (LICS)*, Trento, Italy, July 1999.

[BGNR03] Miquel Bofill, Guillem Godoy, Robert Nieuwenhuis, and Albert Rubio. Paramodulation and Knuth-Bendix Completion with Nontotal and Nonmonotonic Orderings. *Journal of Automated Reasoning*, 30(1):99–120, 2003.

[BR02] Miquel Bofill and Albert Rubio. Well-foundedness is sufficient for completeness of ordered paramodulation. In Andrei Voronkov, editor, *Proceedings of the 18th Conference on Automated Deduction (CADE-18)*, volume 2392 of *LNAI*, Copenhagen, Denmark, July 2002. Springer-Verlag.

[BR04] M. Bofill and A. Rubio. Redundancy notions for paramodulation with non-monotonic orderings. Available at `www.lsi.upc.es/~albert/papers.html`.

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

[KL80] S. Kamin and J.-J. Levy. Two generalizations of the recursive path ordering. Unpublished note, Dept. of Computer Science, Univ. of Illinois, 1980.

[NR01] Robert Nieuwenhuis and Albert Rubio. Paramodulation-based theorem proving. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 7, pages 372–444. Elsevier Science Publishers and MIT Press, 2001.

[Wec91] W. Wechler. *Universal Algebra for Computer Scientists*, volume 25 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1991.