# SOLVING SYMBOLIC ORDERING CONSTRAINTS*

HUBERT COMON†

*CNRS and Laboratoire de Recherche en Informatique*
*Bat. 490, Université de Paris Sud,*
*91405 ORSAY cedex, France.*
*comon@sun8.lri.fr*

Received
Revised

ABSTRACT

We show how to solve boolean combinations of inequations $s > t$ in the Herbrand Universe, assuming that $\geq$ is interpreted as a *lexicographic path ordering* extending a total precedence. In other words, we prove that the existential fragment of the theory of a lexicographic path ordering which extends a total precedence is decidable.

*Keywords*: simplification orderings, ordered strategies, term algebras, constraint solving.

## 1. Introduction

The first order theory of term algebras over a language (or alphabet) with no relational symbol (other than equality) has been shown to be decidable[1,2]. See also Refs 3 and 4. Introducing into the language a binary relational symbol $\geq$ interpreted as the subterm ordering makes the theory undecidable[5]. Venkataraman also shows in the latter paper that the *purely existential fragment of the theory*, i.e. the subset of sentences whose prenex form does not contain $\forall$, is decidable. Venkataraman was concerned with some applications in functional programming which fulfill the interpretation of $\geq$. We are interested in the purely existential fragment of the theory *when $\geq$ is interpreted as a lexicographic path ordering.* Let us briefly consider the motivations for such an interpretation.

*An abstract of this paper appeared in *Proc. IEEE Logic in Computer Science*, Philadelphia, 1990, under the title "Solving inequations in term algebras".

†This research was partly supported by the *Greco de Programmation* and partly by the ESPRIT Basic Research Action COMPASS.

Ordered rewriting and unfailing completion techniques have been introduced in Ref. 6 and successfully applied for deciding (or, in general, semi-deciding) word problems in equational theories. The idea is to replace the relation $\leftrightarrow_E$ (the replacement of equals by equals) with the relation $\leftrightarrow_E^{\succ}$ (ordered rewriting) which consists in an ordered strategy for replacement of equals by equals defined as follows. $u \leftrightarrow_E^{\succ} v$ if $u \leftrightarrow_E v$ and $u \succ v$ for a given simplification ordering $\succeq$ that is total on ground terms (see e.g. Ref. 7 for the missing definitions). In other words, an equation $s = t \in E$ is split into two constrained equations $s > t : s = t$ and $t > s : s = t$. $s = t$ is viewed as (the denotation of) an (infinite) term rewriting system $\{u\sigma \to v\sigma \mid u\sigma \succ v\sigma\} \cup \{v\sigma \to u\sigma \mid v\sigma \succ u\sigma\}$. The word problem is then solved by completion. Adding new equational consequences to $E$ leads to simpler equational proofs than the brute-force replacement of equals by equals[8,9]. The equational consequences are computed by superposition of two equations (the so-called critical pairs): if $u = v$ and $s = t$ are two equations in $E$ and if some non-variable subterm (at position $p$) of $u$ is unifiable with $s$ (with m.g.u. $\sigma$), the pair $(v\sigma, u[t\sigma]_p\sigma)$ is a *critical pair* provided that there exists a ground substitution $\theta$ satisfying $u\sigma\theta \succ v\sigma\theta \wedge s\sigma\theta \succ t\sigma\theta$. The existence of the substitution $\theta$ corresponds exactly to the problem of deciding whether $s\sigma \succ t\sigma \wedge u\sigma \succ v\sigma$ has a solution. Up to now, this problem was unsolved when $\succeq$ is interpreted as a recursive path ordering.

Other applications in the same vein (ordered strategies) are investigated in Ref. 10. For example, the resolution rule is restricted to *ordered resolution* in the following way:

$$\frac{P \vee Q \quad \neg P' \vee R}{Q\sigma \vee R\sigma} \quad \text{If } \sigma = \text{mgu}(P, P') \text{ and } \exists\theta, P\sigma\theta \succ Q\sigma\theta \wedge P\sigma\theta \succ R\sigma\theta.$$

This is a complete strategy (together with ordered factorization, it is a complete set of inference rules), provided that $\succeq$ is total on ground terms. It requires however that one decides whether $P\sigma \succ Q\sigma \wedge P\sigma \succ R\sigma$ is satisfiable in the Herbrand Universe.

For all such applications, $\succeq$ needs to be interpreted as an ordering on ground terms which enjoys the following properties:

1. It is a total ordering on ground terms. (This is required for the completeness of the strategy.)
2. It is monotonic (in the sense of Ref. 7). This is required for handling equality.
3. It is well founded. This is required if we want to incorporate simplification rules. This property is also required for completeness.

There is one well-known ordering on terms which fulfills these three requirements : the *lexicographic path ordering* whose definition is recalled below (see also Ref. 7). This is why we consider such an interpretation of $\geq$ in this paper.

The main result of the paper is to show how to decide whether there is indeed a $\theta$ satisfying $u\sigma\theta \geq_{lpo} v\sigma\theta \wedge s\sigma\theta \geq_{lpo} t\sigma\theta$ (where $\geq_{lpo}$ is the lexicographic path ordering), therefore solving the above questions.

Our result also shows that the ordering on terms with variables defined as $t \succ_* u$ iff, for all ground substitution $\sigma$, $t\sigma \succ u\sigma$ is a decidable simplification ordering when

$\succ$ is a lexicographic path ordering on ground terms. $\succ_*$ generalizes the usual extension of recursive path ordering to terms with variables. This means that more (sometimes strictly more) terms are comparable w.r.t. $\succ_*$ than w.r.t. the recursive path ordering.

Let $\mathcal{T}$ be the theory of term algebra over the relational symbols $=$ and $\geq$, where $\geq$ is interpreted as a lexicographic path ordering. We show the decidability of the purely existential fragment of $\mathcal{T}$. The proof is carried out in three steps. The first step (Sec. 2) consists of the transformation of any quantifier-free formula $\phi$ (i.e. all variables are free) into a *solved form* that has the same set of solutions as $\phi$. In Sec. 3 we reduce the satisfiability of an arbitrary solved form to the satisfiability of some particular problems called *simple systems*. Roughly, a simple system is a formula which defines a total ordering on the terms occurring in it and which is *closed under deduction*. This last property means that, if $\psi$ is a solved form of a simple system $\phi$, then $\psi$ must be a subformula of $\phi$.

In Sec. 4 we show how to reduce the satisfiability of simple systems to the satisfiability of some particular simple systems called *natural simple systems*. Finally, we complete the proof showing that the satisfiability in the Herbrand Universe of a natural simple system is equivalent to the satisfiability of a system of linear inequalities over the integers.

## 2. Inequations Simplification

In the following, $F$ is assumed to be a finite set of *function symbols*. Each function symbol is associated with a non-negative integer $a(f)$ called the *arity of* $f$. When $a(f) = 0$, $f$ is called a *constant*. We assume through this paper that $F$ contains at least one constant. $X$ is a set of *variable symbols*, disjoint from $F$. Terms (or finite trees) over $F$ and $X$ are defined in the usual way (see e.g. Ref. 9 for missing definitions). The set of all terms is denoted by $T(F, X)$. $T(F, \emptyset)$ is simply denoted $T(F)$. Its elements are called *ground terms*.

As in Ref. 9, $t|_p$ denotes the subterm of $t$ at position $p$ and $t[u]_p$ denotes the term $t$ in which $t|_p$ has been replaced with $u$. As usual, $t[u]_p$ is also used to indicate that $u$ is the subterm of $t$ at position $p$. Moreover, when $p$ is not precised, $t[u]$ means that $u$ is a subterm of $t$ at some position.

Substitutions are mappings from $X$ to $T(F, X)$. A substitution $\sigma$ is confused with the (unique) endomorphism of $T(F, X)$ which extends $\sigma$. A *ground substitution* is a substitution $\sigma$ such that $\sigma(X) \subseteq T(F)$.

**Definition 1 (Syntax).** *An* equation *is an expression $s = t$ where $t$ and $t$ are terms in $T(F, X)$. $=$ is assumed to be commutative ($s = t$ is the same equation as $t = s$).*
*An* inequation *is an expression $s > t$ where $s, t \in T(F, X)$.*
*An* inequational problem *is a boolean combination of equations and inequations.*

We use also the following syntactic abbreviations:

- We write sometimes $s < t$ in place of $t > s$
- $\perp$ is the empty disjunction and $\top$ is the empty conjunction

- $s \geq t$ (resp. $t \leq s$) stands for $s > t \vee s = t$.

$\equiv$ denotes the syntactic equality on terms (resp. the syntactic equality of inequational problems). Let $\geq_F$ be a total ordering on $F$. The *lexicographic path ordering* $\geq_{lpo}$ is defined on $T(F, X)$ by $s \geq_{lpo} t \Leftrightarrow s >_{lpo} t \vee s \equiv t$, and $>_{lpo}$ is defined in the following way (see also Ref. 7):

$$s \equiv f(s_1, \ldots, s_n) >_{lpo} g(t_1, \ldots, t_m) \equiv t$$

iff one of the following holds

- $\exists i,\ s_i \geq_{lpo} t$
- $f >_F g$ and $\forall j, s >_{lpo} t_j$
- $f = g$, $(s_1, \ldots, s_n) \gg_{lpo} (t_1, \ldots, t_n)$ and $\forall j,\ s >_{lpo} t_j$

where $(s_1, \ldots, s_n) \gg_{lpo} (t_1, \ldots, t_n)$ iff $\exists j \leq n, \forall i < j, s_i \equiv t_i$ and $s_j >_{lpo} t_j$.

**Definition 2 (Semantics).** *A* solution *of an equation $s = t$ (resp. an inequation $s > t$) is a ground substitution $\sigma$ (i.e. an assignment $\sigma$ from $X$ to $T(F)$) such that $s\sigma \equiv t\sigma$ (resp. $s\sigma >_{lpo} t\sigma$).*

This definition of a solution is extended to inequational problems in a standard way. If $\mathcal{I}$ is an inequational problem, then $\mathcal{S}(\mathcal{I})$ denotes the set of its solutions. The decidability problem we address in this paper is the emptiness of $\mathcal{S}(\mathcal{I})$.

$\geq_{lpo}$ is a total ordering on ground terms (see e.g. Ref. 7). Therefore, $\neg(s > t)$ can be replaced with $s < t \vee s = t$ and $\neg(s = t)$ with $s > t \vee t > s$, without changing the set of solutions of the formula. This is the reason why we assume in the following (without loss of generality) that inequational problems have the following form:

$$\bigvee_{j \in J} s_1 = t_1 \wedge \ldots \wedge s_n = t_n \wedge u_1 > v_1 \wedge \ldots \wedge u_m > v_m.$$

By convention, if $J$ is empty, then the inequational problem is $\perp$ and, if $n = m = 0$ and $J$ is not empty, then the inequational problem is $\top$.

More precisely, let $\rightarrow_{N_1}$ (resp. $\rightarrow_{N_2}$) be the reduction relation defined on inequational problems by the rules of Fig. 1 (resp. 2). Both sets of rules define a canonical term rewriting system modulo the commutativity of $=$ and the associativity and commutativity of $\wedge$ and $\vee$ (see Ref. 9 for definitions). Moreover, the normal form $\mathcal{I}\downarrow_{N_1}\downarrow_{N_2}$ (abbreviated $\mathcal{I}\downarrow_N$) of an inequational formula $\mathcal{I}$ has the same set of solutions as $\mathcal{I}$. We may therefore restrict our attention to irreducible problems w.r.t. $\rightarrow_{N_1} \cup \rightarrow_{N_2}$ and call them *disjunctive normal forms.*[a]

**Definition 3 (Solved Forms).** *A* solved form *is either $\top$, $\perp$ or a formula*

$$x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge u_1 > v_1 \wedge \ldots \wedge u_m > v_m,$$

*where*

[a] Let us emphasize that, with our definition, there is no identity $s = s$ in a disjunctive normal form: $s = s \equiv \top$.

$$
\begin{array}{rcl}
\neg(s=t) & \rightarrow & s>t \vee t>s \\
\neg(s>t) & \rightarrow & t>s \vee s=t \\
\neg\top & \rightarrow & \bot \\
\neg\bot & \rightarrow & \top \\
\neg(a \vee b) & \rightarrow & \neg a \wedge \neg b \\
\neg(a \wedge b) & \rightarrow & \neg a \vee \neg b
\end{array}
$$

Fig. 1. Elimination of negation from inequational problems.

$$
\begin{array}{rcl}
s=s & \rightarrow & \top \\
s>s & \rightarrow & \bot \\
s<t & \rightarrow & t>s \\
(a \vee b) \wedge c & \rightarrow & (a \wedge c) \vee (b \wedge c) \\
a \wedge a & \rightarrow & a \\
a \vee a & \rightarrow & a \\
a \vee \top & \rightarrow & \top \\
a \wedge \top & \rightarrow & a \\
a \vee \bot & \rightarrow & a \\
a \wedge \bot & \rightarrow & \bot
\end{array}
$$

Fig. 2. Normalization of inequational problems.

- *$x_1, \ldots, x_n$ are variables occurring only once in the formula*
- *for each $i \in \{1, \ldots, m\}$, $u_i$ or $v_i$ is a variable*
- *for each index $i \in \{1, \ldots, m\}$, $u_i$ is not a subterm of $v_i$ nor $v_i$ of $u_i$*

We now describe a set of rules $\mathcal{R}$ for the transformation of inequational formulas. The corresponding reduction relation is written out $\Rightarrow_{\mathcal{R}}$. $\mathcal{R}$ is called *correct* if $\phi \Rightarrow_{\mathcal{R}} \phi'$ implies that $\phi$ and $\phi'$ have the same solutions. Given a set of solved forms (as above) $\mathcal{R}$ is called *complete* if any normal form for $\Rightarrow_{\mathcal{R}}^*$ is a solved form.

Each rule in Fig. 3 is followed by a (possibly empty) condition. This defines a class of algorithms by choosing any sequence of reductions that fulfills the conditions. The rules apply to disjunctive normal forms of problems. Therefore, we assume that a normalization (w.r.t. $\rightarrow_N$) is performed after each reduction with $\Rightarrow$. (This must be kept in mind when proving termination).

**Proposition 1** . *The rules given in Fig. 3 are correct, complete and terminating.*

**Proof.** Correctness is a direct consequence of the definition of $>_{lpo}$: assume that $s \equiv f(s_1, \ldots, s_n)$ and $t \equiv g(t_1, \ldots, t_m)$ are two ground terms. Then $s >_{lpo} t$ iff one of the following holds:

1. $f >_F g$ and either $\exists i, s_i \geq_{lpo} t$ or $\forall i, s >_{lpo} t_i$. However, if $s_i \geq_{lpo} t$, then $s >_{lpo} t_i$. Therefore, the former case is useless. This corresponds to the rule $(D_2)$.
2. $g >_F f$, then one of the $s_i$'s must be larger than $t$. This corresponds to rule $(D_3)$.
3. $f = g$, then, either one of the $s_i$'s is larger than $t$ or $(s_1, \ldots, s_n) \gg_{lpo} (t_1, \ldots, t_n)$. This corresponds to rule $(D_4)$.

The correctness of other rules is obvious.

Completeness is easy to check when the system terminates. Let us prove termination. Consider the following interpretation functions:

- $\phi_1(s_1 = t_1 \wedge \ldots \wedge s_n = t_n \wedge u_1 > v_1 \wedge \ldots \wedge u_m > v_m)$ is the multiset of multisets of natural numbers:
$$\{\{|s_1|, |t_1|\}, \ldots, \{|s_n|, |t_n|\}, \{|u_1|, |v_1|\}, \ldots, \{|u_m|, |v_m|\}\}$$
where $|s|$ is the number of function symbols and variables occurring in $s$ (also called the *size* of $s$). Such multisets are ordered by the usual multiset extensions of orderings (see e.g. Ref. 11).
- $\phi_2(s_1 = t_1 \wedge \ldots \wedge s_n = t_n \wedge u_1 > v_1 \wedge \ldots \wedge u_m > v_m)$ is the number of *unsolved variables* in the system. A variable $x$ is solved in such a system if $x$ is a member of an equation and $x$ occurs only once in the system.
- $\phi(\bigvee_{j \in J} c_j)$, where $c_j$ is a conjunction of equations and inequations, is the multiset of pairs $(\phi_2(c_j), \phi_1(c_j))$. Such interpretations are ordered using the multiset extension of the lexicographic ordering on pairs.

**Equality Rules**

$$(D_1) \quad f(v_1,\ldots,v_n) = f(u_1,\ldots,u_n) \quad \Rightarrow \quad v_1 = u_1 \wedge \ldots \wedge v_n = u_n$$

$$(C_1) \quad f(v_1,\ldots,v_n) = g(u_1,\ldots,u_m) \quad \Rightarrow \quad \perp \qquad \text{If } f \neq g$$

$$(R) \quad x = t \wedge P \quad \Rightarrow \quad x = t \wedge P\{x \mapsto t\}$$

If $x$ is a variable, $x \notin Var(t)$, $P$ is a conjunction of equations and inequations, $x \in Var(P)$ and, if $t$ is a variable, then $t \in Var(P)$.

$$(O_1) \quad s = t[s]_p \quad \Rightarrow \quad \perp \qquad \text{If } p \neq \Lambda.$$

**Inequality Rules**

$$\begin{aligned}(D_2) \quad f(v_1,\ldots,v_n) > g(u_1,\ldots,u_m) \ \Rightarrow \\ f(v_1,\ldots,v_n) > u_1 \wedge \ldots f(v_1,\ldots,v_n) > u_m\end{aligned}$$

If $f >_F g$

$$\begin{aligned}(D_3) \quad f(v_1,\ldots,v_n) > g(u_1,\ldots,u_m) \ \Rightarrow \\ v_1 \geq g(u_1,\ldots,u_m) \vee \ldots v_n \geq g(u_1,\ldots,u_m)\end{aligned}$$

If $g >_F f$

$$\begin{aligned}(D_4) \quad f(v_1,\ldots,v_n) > f(u_1,\ldots,u_n) \ \Rightarrow \\ & (v_1 > u_1 \wedge f(v_1,\ldots,v_n) > u_2 \wedge \ldots f(v_1,\ldots,v_n) > u_n) \\ & \vee (v_1 = u_1 \wedge v_2 > u_2 \wedge \ldots f(v_1,\ldots,v_n) > u_n) \\ & \vee \ \ldots \\ & \vee (v_1 = u_1 \wedge v_2 = u_2 \wedge \ldots v_n > u_n) \\ & \vee v_1 \geq f(u_1,\ldots,u_n) \vee \ldots \vee v_n \geq f(u_1,\ldots,u_n)\end{aligned}$$

$$(O_2) \quad t[s]_p > s \quad \Rightarrow \quad \top \qquad \text{If } p \neq \Lambda.$$

$$(O_3) \quad s > t[s] \quad \Rightarrow \quad \perp$$

$$\begin{aligned}(T_1) \quad s > t \wedge t > s \quad \Rightarrow \quad \perp \\ (T_2) \quad s = t \wedge s > t \quad \Rightarrow \quad \perp\end{aligned}$$

Fig. 3. Transformation rules.

We prove actually that $\phi$ is strictly decreasing by application of any rule to an inequational problem. Assume that $\mathcal{I} \equiv c \vee \bigvee_{j \in J} c_j$, and $c \Rightarrow \bigvee_{j \in J'} c'_j$. Then, we have to prove that, for every $j \in J'$, either $\phi_2(c'_j) < \phi_2(c)$ or $\phi_2(c'_j) = \phi_2(c)$ and $\phi_1(c'_j) < \phi_1(c)$.

Actually, for every rule and every $j$, $\phi_2(c'_j) \leq \phi_2(c)$ because there is no rule which can turn a solved variable into an unsolved one, except if the resulting formula is $\top$ or $\bot$. Assume now that $\phi_2(c'_j) = \phi_2(c)$ for some $j$. Note that this excludes the replacement rule $(R)$ for which the number of unsolved variables is strictly decreasing. It is easy to check the strict decreasingness of $\phi_1$. Let us show it for e.g. $(D_3)$:

$$\begin{aligned} c \equiv c' \wedge f(v_1, \ldots, v_n) > g(u_1, \ldots, u_m) \Rightarrow & \\ & (c' \wedge v_1 = g(u_1, \ldots, u_m)) \vee \ldots \vee (c' \wedge v_m = g(u_1, \ldots, u_m)) \\ & \vee (c' \wedge v_1 > g(u_1, \ldots, u_m)) \vee \ldots \vee (c' \wedge v_n > g(u_1, \ldots, u_m)). \end{aligned}$$

For each $j$, $c'_j$ is either some $c' \wedge v_i = g(u_1, \ldots, u_m)$ or some $c' \wedge v_i > g(u_1, \ldots, u_m)$. In both cases,

$$\phi_1(c) = \{a_1, \ldots, a_k, \{1 + b_1 + \ldots + b_n, c\}\},$$

where, for each $i$, $b_i = |v_i|$ and $c = |g(u_1, \ldots, u_m)|$ and

$$\phi_1(c'_j) = \{a_1, \ldots, a_k, \{b_i, c\}\},$$

for some $i$. By definition of the multiset ordering, $\phi_1(c'_j) < \phi_1(c)$.

This proves that $\phi$ is strictly decreasing by application of any rule. Since $\phi$ interprets the inequational problems in a well-founded domain, this proves termination. $\square$

**Example 1**. Assume that $F' = F \cup \{h, g\}$ (where $h, g \notin F$) with $h >_F g >_F f$. Then,

$$h(u_1, u_2) > g(h(u_1, v_2), h(v_1, u_2)) \;\Rightarrow^*_{\mathcal{R}}\; u_1 > v_1 \wedge u_2 > v_2.$$

This shows that solving a conjunction of inequations is equivalent to solving a single inequation w.r.t. another set of function symbols.

The reduction relation $\Rightarrow_{\mathcal{R}}$ is not sufficient for deciding the existence of a solution. Indeed, there are irreducible inequational problems that are different from $\bot$ but do not have any solution.

**Example 2**. Let $F = \{\ldots, s, 0\}$ with $\ldots >_F s >_F 0$. The following problem:

$$s(x) > y \wedge y > x$$

has no solution since, for every ground term $x$, there is no term between $x$ and $s(x)$

The above example shows that our rules system is not sufficient. It suggests use of the following rules:

$$\begin{array}{rcl} v > u & \rightsquigarrow & v = succ(u) \vee v > succ(u) \\ succ(u) > v & \rightsquigarrow & v = u \vee u > v \end{array}$$

If $succ(u)$ is a term such that, for all ground substitution $\sigma$, there is no term between $succ(u)\sigma$ and $u\sigma$. (We say that $succ(u)\sigma$ is *the successor* of $u\sigma$).

Unfortunately, there are two problems with these rules. First, they do not terminate because we can derive an infinite sequence $v > succ^n(u)$ (there may be a "gap" between $v$ and $u$). Therefore, we have to find in which situations they should be used. Secondly they are not complete. Indeed, a term $u$ may have some instances that are successor terms and some instances that are not successor terms.

**Example 3**. Assume $F = \{g >_F f >_F 1 >_F 0\}$, where $f$ is a binary function symbol. Then, $f(x, y)$ has some instances which are successor terms ($f(0, g(0))$ is the successor of $g(0)$) and some instances which are not successor terms: $f(1, 0)$ is a "limit ordinal":

$$0 < 1 < f(0,0) < f(0,1) < f(0, f(0,0)) < f(0, f(0,1)) \ldots < f(1,0) \ldots$$

Even worse, it may happen that all instances of a term $v$ are successor terms but they cannot be written as successors of instances of a single term $u$.

**Example 4**. Using the same ordering $\geq_F$ as in Example 3, $f(0, x)$ is a successor term for every ground term $x$. But $f(0,0) = succ(1)$, whereas $f(0,t) = succ(t)$ if $t \geq_{lpo} f(1,0)$ (See Sec. 4 for more details). Then, for example, the problem $f(0,x) > y \wedge x > f(1,0) \wedge y > x$ has no solution.

This shows that we have to study more deeply the successor function in order to derive the right (terminating) rules. This will be achieved in Sec. 4. Before that, let us simplify the problem by reducing the satisfiability of solved forms to the satisfiability of some particular formulas.

## 3. From Solved Forms to Simple Systems

In this second step, we show how to reduce the satisfiability of inequational problems to the satisfiability of *simple systems*.

The basic idea is to perform at once all possible identifications between terms occurring in $\mathcal{I}$. Then, one need no longer consider equalities: we may then assume that all terms occurring in $\mathcal{I}$ are distinct and may therefore be totally ordered.

**Example 5**. Let $\mathcal{I} \equiv f(y, 0) > x \wedge y > f(0, x)$. We sketch our method on this simple example. The possible identifications are $x = 0$, $x = y$, $x = f(y, 0)$, $y = 0$, $y = f(0, x)$ and all combinations of them.

Consider, for example, the identification $x = 0$. We get the system

$$f(y,0) > 0 \wedge y > f(0,0).$$

Now the terms occurring in the system are assumed to be distinct and hence can be totally ordered. This leads to the systems

$$f(y,0) > y > 0 > f(0,0); \quad f(y,0) > y > f(0,0) > 0, \ldots.$$

The first system can be removed since $0 > f(0,0) \Rightarrow^*_{\mathcal{R}} \perp$. The second problem is a simple system.

The section is organized as follows: we first define simple systems (3.1), then perform all possible identifications (3.2) and finally consider all total orderings on the subterms which do not lead to a contradiction, using $\mathcal{R}$ (3.3).

### *3.1. Definition of Simple Systems*

A *system* is a conjunction of equations and inequations. It can be considered either as a formula or a set of equations and inequations. If $\mathcal{I}$ is a system, then $Sub(\mathcal{I})$ denotes the set of all (sub)terms of all terms $t$ that occur as a member of an equation or an inequation in $\mathcal{I}$.

A *simple system* is a system $\mathcal{I}$ satisfying the following properties:

- There exists a finite set of terms $\{t_1, \ldots, t_n\}$ such that

$$\mathcal{I} \equiv \bigwedge_{1 \le i < j \le n} t_i > t_j.$$

- If $\mathcal{I} \Rightarrow_{\mathcal{R}} \bigvee_{j \in J} c_j$, then some $c_j$ is a subformula of $\mathcal{I}$ (i.e. there exists a formula $c$ such that $\mathcal{I} \equiv c \wedge c_j$). This means in particular that $J$ cannot be empty.
- Every subterm of a term $t_i$ is some term $t_j$.

Such systems will be written in the following way

$$t_1 > \ldots > t_n.$$

Note that a simple system, by definition, cannot be reduced to $\perp$ using $\mathcal{R}$, because of the second condition above. In particular, a simple system cannot contain any inequation $s > t[s]$.

Moreover, note that the second condition above is equivalent to the (stronger) following property:

> If $\mathcal{I}'$ is a subsystem of $\mathcal{I}$ and $\mathcal{I}' \Rightarrow^*_{\mathcal{R}} \bigvee_{j \in J} c_j$, then some $c_{j_0}$ is a subsystem of $\mathcal{I}$.

In the following we will use as well this stronger version of the second condition.

### *3.2. Identifications*

A variable $x$ that occurs in a system $\mathcal{I}$ is *solved in* $\mathcal{I}$ if $x$ occurs only once in $\mathcal{I}$ and there is an equation $x = t$ in $\mathcal{I}$. We consider the following rule (identification)

$$(Id) \;\; \mathcal{I} \rightarrow_{Id} x = t \wedge \mathcal{I}\{x \mapsto t\},$$

> if $x \in Var(\mathcal{I}) - Var(t)$ is not solved in $\mathcal{I}$ and $t \in Sub(\mathcal{I})$ and $t$ is not a solved variable.

The reduction relation $\rightarrow_{Id}$ is terminating because each application of the rule decreases the set of unsolved variables in the system. Therefore, the set $G(\mathcal{I})$ of systems $\mathcal{I}'$ (in disjunctive normal form) such that $\mathcal{I} \rightarrow^*_{Id} \mathcal{I}'$ is finite.

The *equational part* of a system $\mathcal{I}$ is the conjunction of all equations occurring in $\mathcal{I}$. The *inequational part* $IP(\mathcal{I})$ is the conjunction of inequations occurring in $\mathcal{I}$. Let

$$H(\mathcal{I}) = \{\mathcal{I}' \mid \exists \mathcal{I}'' \in G(\mathcal{I}), \mathcal{I}' = IP(\mathcal{I}'')\}.$$

**Example 6**. Let $f >_F 0$ and $\mathcal{I} \equiv f(y,0) > x \wedge y > f(0,x)$. Then,

$$\begin{array}{rll} H(\mathcal{I}) = \{ & f(y,0) > x \wedge y > f(0,x); & f(y,0) > 0 \wedge y > f(0,0); \\ & f(y,0) > y \wedge y > f(0,y); & \perp; \\ & f(0,0) > x \wedge 0 > f(0,x); & f(0,0) > 0 \wedge 0 > f(0,0); \\ & \} & \end{array}$$

**Lemma 1** . *$H(\mathcal{I})$ is finite.*

**Proof.** This follows the termination property of $\rightarrow_{Id}$. $\square$

Now, because all possible identifications have been considered, we may assume that all terms occurring in a formula $\mathcal{I}' \in H(\mathcal{I})$ are distinct:

**Lemma 2** . *$\mathcal{I}$ has a solution iff there is an inequational problem $\mathcal{I}' \in H(\mathcal{I})$ which has a solution $\sigma$ such that*

$$\forall s, t \in Sub(\mathcal{I}'), s\sigma \not\equiv t\sigma.$$

**Proof.** If $\sigma$ is a solution of some $\mathcal{I}' \in H(\mathcal{I})$, then, by construction, there is an $\mathcal{I}'' \equiv x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge \mathcal{I}'$ where $x_1, \ldots, x_n$ are solved in $\mathcal{I}''$ and $\mathcal{I} \Rightarrow^*_{Id} \mathcal{I}''$. Then $\{x_1 \mapsto t_1; \ldots x_n \mapsto t_n\}\sigma$ is a solution of $\mathcal{I}$.

Conversely, if $\sigma$ is a solution of $\mathcal{I}$, then we construct $G_\sigma(\mathcal{I})$ as follows:

- $\mathcal{I}_0 \equiv \mathcal{I}$.
- If there is a variable $x \in Var(\mathcal{I}_n)$ and a term $t \in Sub(\mathcal{I}_n)$ such that $x\sigma \equiv t\sigma$ and $\mathcal{I}_n \rightarrow_{Id} x = t \wedge \mathcal{I}_n\{x \mapsto t\}$, then $\mathcal{I}_{n+1} \equiv x = t \wedge \mathcal{I}_n\{x \mapsto t\}$.
- Otherwise, $G_\sigma(\mathcal{I}) = \mathcal{I}_n$.

The above construction terminates because $\rightarrow_{Id}$ terminates, and, by definition, $IP(G_{\sigma(\mathcal{I})}) \in H(\mathcal{I})$. Moreover, by construction, we have the following properties:

- $\sigma$ is a solution of $G_\sigma(\mathcal{I})$ ($\sigma$ is a solution of each $\mathcal{I}_n$).
- If $x \in Var(IP(G_\sigma(\mathcal{I})))$ and $t \in Sub(IP(G_\sigma(\mathcal{I})))$, then $x\sigma \not\equiv t\sigma$. $\square$

*3.3. Considering all Compatible Total Orderings*

For every system $\mathcal{I}$, let $K(\mathcal{I})$ be the set of all total orderings $\succ$ on $Sub(\mathcal{I})$ compatible with the inequations in $\mathcal{I}$ i.e. such that $s > t \in \mathcal{I} \Rightarrow s \succ t$. Then, if $\mathcal{I}$ is a system, let

$$S(\mathcal{I}) = \{\bigwedge_{s \succ t} s > t \ \mid \succ \in K(\mathcal{J}), \mathcal{J} \in H(\mathcal{I})\}.$$

Finally, let $\mathcal{D}(\mathcal{I})$ be the set of systems $\mathcal{I}'$ in $S(\mathcal{I})$ that cannot be reduced to $\perp$ using $\Rightarrow^*_{\mathcal{R}}$.

The following lemmas show that the satisfiability of an inequational problem reduces to the satisfiability of some system in $\mathcal{D}(\mathcal{I})$, which is a finite set of simple systems.

**Lemma 3** . *A conjunction of inequations $\mathcal{I}$ has a solution iff there is some system in $\mathcal{D}(\mathcal{I})$ that has a solution.*

**Proof.** If some system in $\mathcal{D}(\mathcal{I})$ has a solution $\sigma$, then $\sigma$ is also the solution of some $\mathcal{I}' \in H(\mathcal{I})$ and, by Lemma 2, $\mathcal{I}$ has a solution.

Conversely, assume that $\sigma$ is a solution of $\mathcal{I}$. Then, by Lemma 2, $\sigma$ is a solution of some $\mathcal{I}'' \in H(\mathcal{I})$ and is injective on $Sub(\mathcal{I}'')$. Therefore, $\sigma$ is a solution of some system $\mathcal{I}' \in S(\mathcal{I})$ : it is sufficient to choose $\succ$ as follows:

$$s \succ t \Leftrightarrow s\sigma >_{lpo} t\sigma.$$

This indeed defines a total ordering on $Sub(\mathcal{I}'')$ because, if $s\sigma \equiv t\sigma$ and $s \not\equiv t$, there is a variable $x$ in $Var(s,t)$ and a subterm $u$ of $s$ or $t$ such that $x\sigma \equiv u\sigma$, which contradicts the injective property of $\sigma$. Now, $\mathcal{I}'$ cannot be reduced to $\perp$ (since it has a solution) : $\mathcal{I}' \in \mathcal{D}(\mathcal{I})$. $\square$

**Lemma 4** . *$\mathcal{D}(\mathcal{I})$ is a finite computable set of simple systems.*

**Proof.** We only have to prove that, if $\mathcal{I}' \in \mathcal{D}(\mathcal{I})$ and $\mathcal{I}' \Rightarrow_{\mathcal{R}} \bigvee_{j \in J} c_j$, then some $c_j$ is a subformula of $\mathcal{I}'$. Since $\mathcal{I}'$ is a conjunction of inequations, only $(D_2), (D_3), (D_4), (O_2), (O_3), (T_1), (T_2)$ may apply. Since $\mathcal{I}' \not\Rightarrow^*_{\mathcal{R}} \perp$, $(O_3), (T_1), (T_2)$ cannot apply. Moreover, if the rule replaces an inequation of $\mathcal{I}'$ with $\top$,[b] then $J$ contains a single element (assume it is 1). $c_1$ is obviously a subformula of $\mathcal{I}'$. We exclude now this case. This means in particular that only the rules $(D_2), (D_3), (D_4)$ have still to be considered.

Assume now that $\mathcal{I} \mapsto_{D_2, D_3, D_4} \bigvee_{i \in J} c_i$. Let $i_0$ be an index $i_0 \in J$ such that $c_{i_0} \not\Rightarrow^*_{\mathcal{R}} \perp$. Such an index does exist since $\mathcal{I}' \not\Rightarrow^*_{\mathcal{R}} \perp$. $Sub(c_{i_0}) \subseteq Sub(\mathcal{I}')$ because each term occurring in the right-hand side of a rule $D_i$ is a subterm of some term occurring in the left-hand side. On the other hand, if $s, t \in Sub(\mathcal{I}')$ are two distinct terms, then either $s > t$ or $t > s$ occurs in $\mathcal{I}'$, by construction. Then, for every equation $s = t$ and every inequation $s > t$ occurring in $c_{i_0}$, either $s > t$ or $t > s$ occurs in $\mathcal{I}'$. This means that

[b] This occurs if the rule $(O_2)$ is used, but also in other situations, for example, with the rule $(D_3)$ when $t_i \equiv g(u_1, \ldots, u_m)$. Indeed, in such a case, the rule produces an identity $s = s$ which is immediately reduced to $\top$ by the normalization of formulas.

1. $c_{i_0}$ contains no equation. (Otherwise, it can be reduced to $\perp$ by $(T_2)$).

2. if $s > t \in c_{i_0}$, then $s > t \in \mathcal{I}'$. (Otherwise, $c_{i_0}$ contains both inequations $s > t$ and $t > s$. Then it could be reduced to $\perp$ by $(T_1)$.)

This shows that $c_{i_0}$ is a subformula of $\mathcal{I}'$. $\square$

From the previous lemmas, it is sufficient to study simple systems. This is what we are going to do in the next section.

## 4. Satisfiability of Simple Systems

We show here how to decide the satisfiability of simple systems. Once again, we split the satisfiability problem in three steps. First, we establish some technical results about $\geq_{lpo}$. Then we show how to solve a particular kind of simple systems: the *natural simple systems*. Finally, we reduce the satisfiability of a simple system to the satisfiability of finitely many natural simple systems.

### *4.1. Successor of a Ground Term*

The ordering $\geq_F$ is assumed to be total. We assume moreover that $F$ contains at least one non-constant function symbol. (If $F$ is a set of constants, then $T(F)$ is finite and it is easy to find all solutions of an inequational problem, trying all possible ground instances of the variables.)

Let $0$ be the least constant symbol and $f$ be the least non-constant function symbol. $C$ is the set of constant function symbols that are either $0$ or smaller (w.r.t. $\geq_F$) than any non-constant function symbol. Let $C = \{c_1 >_F \ldots >_F c_n = 0\}$.

Some terms play a special role w.r.t. the successor function in $T(F)$ (there is a "discontinuity" as shown in Sec. 2). Let $\mathcal{N}$ be the least set of terms solution of the equation $\mathcal{N} = C \cup f(0, \ldots, 0, \mathcal{N})$. We also denote by $\mathcal{N}(X)$ the least solution of

$$\mathcal{N}(X) = C \cup X \cup f(0, \ldots, 0, \mathcal{N}(X)).$$

Note that any term in $\mathcal{N}(X)$ contains at most one occurrence of a variable.

$succ(t)$ denotes the *successor of* $t \in T(F)$ defined as follows:

- $succ(c_{i+1}) = c_i$.
- $succ(c_1) = f(0, \ldots, 0)$.
- If $t \equiv f(0, \ldots, 0, t') \in \mathcal{N}$, then $succ(t) = f(0, \ldots, 0, succ(t'))$.
- In all other cases, $succ(t) = f(0, \ldots, 0, t)$.

**Lemma 5** . *Let $s, t$ be two ground terms. $s >_{lpo} t$ iff $s \geq_{lpo} succ(t)$.*

In particular, there is no term between $t$ and $succ(t)$.

**Proof.** $succ(t) >_{lpo} t$ is straightforward. We only have to prove that $s >_{lpo} t \Rightarrow s \geq_{lpo} succ(t)$. We are going to prove this implication by induction on the depth of $t$.

If $t \in C$, this is straightforward. If $t$ is a constant and $t \notin C$, then $s >_{lpo} t$ implies that there is a subterm $u \equiv g(u_1, \ldots, u_m)$ in $s$ such that $g \geq_F t$. On the other hand, $t >_F f$ since $t \notin C$. Therefore, $g(u_1, \ldots, u_m) >_{lpo} t \Rightarrow g(u_1, \ldots, u_m) >_{lpo} f(0, \ldots, 0, t) \equiv succ(t)$. Finally, if $g = t$, then $u$ is a strict subterm of $s$ and the top symbol of $s$ is greater than $f$, which shows again that $s \geq_{lpo} f(0, \ldots, 0, t) \equiv succ(t)$.

Assume now that the lemma holds up to a certain depth. As above, $s >_{lpo} t \equiv g(t_1, \ldots, t_n)$ implies that

- either there is some subterm $u$ in $s$ such that $u >_{lpo} t$, $u \equiv h(u_1, \ldots, u_m)$, $h \geq_F g$ and no proper subterm of $u$ has this property
- or $t$ is a proper subterm of $s$ and we are not in the above case.

We investigate now a number of cases

**Case 1.** $h >_F g$ and $t \notin \mathcal{N}$.
Then $u >_{lpo} t$ and $u >_{lpo} 0$ implies $u >_{lpo} f(0, \ldots, 0, t)$ since $h >_F f$, by definition of $>_{lpo}$. Therefore, $u >_{lpo} succ(t)$.

**Case 2.** $h >_F g = f$ and $t \in \mathcal{N}$.
Either $t$ is a constant and the result is straightforward or $t \equiv f(0, \ldots, 0, t')$. Then $u \geq_{lpo} succ(t')$ by induction hypothesis. But $u$ cannot equal $succ(t')$ since the top symbol of $u$ is $h$ and the top symbol of $succ(t')$ is $f$, by definition of $succ$. Therefore, $u >_{lpo} succ(t')$. Now, $h >_F f$ implies $u >_{lpo} f(0, \ldots, 0, succ(t')) \equiv succ(t)$.

**Case 3.** $h = g >_F f$.
In such a case, $t \notin \mathcal{N}$ and $h >_F f$ implies $u >_{lpo} f(0, \ldots, 0, t) = succ(t)$ by definition of $>_{lpo}$.

**Case 4.** $h = g = f$, $u_i \not\equiv 0$ for some $i < n$ and $t \notin \mathcal{N}$.
By definition of $>_{lpo}$, $u > t$ and $u_i > 0$ for $i < n$ implies $f(u_1, \ldots, u_n) >_{lpo} f(0, \ldots, 0, t)$.

**Case 5.** $h = g = f$, $u_i \not\equiv 0$ for some $i$ and $t \in \mathcal{N}$.
By induction hypothesis, $u >_{lpo} t_n$ implies $u \geq succ(t_n)$ and, since $u_i \not\equiv 0$, $u$ cannot equal $succ(t_n)$. Therefore, by definition of $>_{lpo}$, $u > f(0, \ldots, 0, succ(t_n))$.

**Case 6.** $u \equiv f(0, \ldots, 0, u_n)$.
Since $u_n \not>_{lpo} t$ and $u >_{lpo} t$, $t$ must equal $f(0, \ldots, 0, t_n)$ and $u_n >_{lpo} t_n$. By induction hypothesis, $u_n \geq succ(t_n)$. Now, either $t \in \mathcal{N}$ and obviously $u >_{lpo} succ(t)$ or $t \notin \mathcal{N}$. In the latter case, $t_n \notin \mathcal{N}$ and $succ(t_n) \equiv t$. Thus, $u_n \geq_{lpo} t$ and therefore $u \geq_{lpo} succ(t)$.

**Case 7.** $t$ is a proper subterm of $s$ and $s \equiv h(s_1, \ldots, s_n)$ satisfies $g >_F h$.
Then $g \neq f$ and $h \notin C$ (and therefore $t \notin \mathcal{N}$). Now, either $h = f$ and $s \geq_{lpo} succ(t)$ follows from $s_1 \geq_{lpo} 0, \ldots, s_n \geq_{lpo} 0$ and $s_i \geq_{lpo} t$ for some $i$, or else $h >_F f$ and the conclusion follows from $s >_{lpo} 0$ and $s >_{lpo} t$. $\square$

Lemma 5 actually states precisely the relationship between ground terms and ordinal numbers: *succ* is the *successor* function.

**Example 7.** Let us come back to Example 3: $g >_F f >_F 1 >_F 0$. Listing the terms in increasing order, we find (using classical ordinal notations):

| $0$ | $1$ | $2$ | $3$ | $4$ | ... |
|---|---|---|---|---|---|
| $0$ | $1$ | $f(0,0)$ | $f(0,1)$ | $f(0,f(0,0))$ | ... |

| $\omega$ | $\omega+1$ | $\omega+2$ | ... |
|---|---|---|---|
| $f(1,0)$ | $f(0,f(1,0))$ | $f(0,f(0,f(1,0)))$ | ... |

| $\omega*2$ | $\omega*2+1$ | ... | $\omega^2$ | $\omega^2+1$ | ... |
|---|---|---|---|---|---|
| $f(1,1)$ | $f(0,f(1,1))$ | ... | $f(1,f(1,0))$ | $f(0,f(1,f(1,0)))$ | ... |

| $\omega^\omega$ | ... | $\omega^{\omega^\omega}$ | ... | $\epsilon_0$ | ... |
|---|---|---|---|---|---|
| $f(f(0,0),0)$ | ... | $f(f(1,0),0)$ | ... | $f(f(f(0,0),0),0)$ | ... |

(See Ref. 12 for a description of the order type of recursive path orderings.)

The set of terms that have indeed a predecessor will be useful. Let $ST$ be the subset of $T(F)$ defined by:

$$ST = \{t \in T(F) \mid \exists u \in T(F), t \equiv succ(u)\}.$$

**Lemma 6** . *We have the following properties of* $ST, \mathcal{N}$*:*

- $ST \cup \{0\} = \mathcal{N} \cup \{f(0,\ldots,0,u) \mid u \in T(F)\}$.
- $\forall t \in T(F), succ(t) \neq f(0,\ldots,0,t) \Rightarrow t \in \mathcal{N}$.
- $\mathcal{N}$ *is order-isomorphic to the natural numbers.*
- *Every term in* $T(F) - \mathcal{N}$ *is greater (w.r.t.* $\geq_{lpo}$*) than every term in* $\mathcal{N}$*.*

**Proof.** The first property follows from the definition of *succ*. The second property follows from the definition of *succ* by induction on the depth of $t$. The third property follows from Lemma 5 since $\mathcal{N}$ is the set of $succ^n(0)$ for every natural number $n$. The fourth proposition is a consequence of the third one. □

Extending the previous notation, $ST_\exists(X)$ will be the set of terms that have an instance in $ST$ and $ST_\forall(X)$ the set of terms whose all ground instances are in $ST$:

**Lemma 7** .

$$ST_\exists(X) \cup \{0\} = \mathcal{N}(X) \cup \{f(y_1,\ldots,y_k,u) \mid \forall i, y_i \in X \cup \{0\},\ u \in T(F,X)\}$$

*and*

$$ST_\forall(X) \cup \{0\} = \mathcal{N} \cup \{f(0,\ldots,0,u) \mid u \in T(F,X)\}.$$

**Proof.** Let $A$ be the right-hand side of the first equality in the lemma. If $t \in A$, then, replacing all the variables of $t$ by $0$ leads to a term in $ST \cup \{0\}$ by Lemma 6.

Therefore, $A \subseteq ST_{\exists}(X) \cup \{0\}$. Conversely, replacing any subterm of $t \in ST \cup \{0\}$ with a variable gives a term in $A$.

If $u \in T(F,X)$, then, for every substitution $\sigma$, $f(0,\ldots,0,u)\sigma \in ST$ by Lemma 6. Conversely, if $t \in ST_{\forall}(X)$, then every ground instance of $t$ is either in $C$ or has the form $f(0,\ldots,0,v)$ by Lemma 5. This means that $t$ is either in $\mathcal{N}$ or has the form $f(0,\ldots,0,u)$ where $u \in T(F,X)$. □

**Example 8**. In the previous example, we find

$$T(F) - ST \supseteq \{f(1,0), f(1,1), f(1,f(1,0)), f(f(0,0),0), f(f(1,0),0), g(0)\}$$

$$\begin{aligned} T(F,X) - ST_{\exists}(X) = \{0\} &\cup \{f(1,t) \mid t \in T(F,X)\} \ \cup \ \{g(t) \mid t \in T(F,X)\} \\ &\cup \{f(f(t_1,t_2),t_3) \mid t_1,t_2,t_3 \in T(F,X)\} \\ &\cup \{f(g(t_1),t_2) \mid t_1,t_2 \in T(F,X)\} \end{aligned}$$

*4.2. Natural Simple Systems*

A *natural simple system* is a simple system $t_1 > \ldots > t_n$ such that, for every $i$, $t_i \in \mathcal{N}(X)$. The following lemma shows that natural simple systems can easily be solved. Also, it will be used for reducing simple systems to natural simple systems.

**Lemma 8** . *Let $\mathcal{I} \equiv t_1 > \ldots > t_n$ be a natural simple system. Then $\mathcal{I}$ has a solution iff it has a solution $\sigma$ such that $\forall x \in Var(\mathcal{I}), x\sigma \in \mathcal{N}$.*

**Proof.** Assume that $\sigma$ is a minimal solution of $\mathcal{I}$. i.e. there is no solution $\theta$ other than $\sigma$ such that, for all variable $x$, $x\sigma \geq_{lpo} x\theta$ (such a $\sigma$ exists because $\geq_{lpo}$ is well-founded). Suppose now that some variable $x$ satisfies $x\sigma \notin \mathcal{N}$. We are going to derive a contradiction.

Let $\mathcal{I} \equiv \ldots t_k > t_{k+1} \ldots$ such that $t_k\sigma \notin \mathcal{N}$ and $t_{k+1}\sigma \in \mathcal{N}$. (It is always possible to assume such a situation, maybe by adding $t_n > 0$ when $k = n$). Each $x_i\sigma$ can be viewed as an ordinal; in order to get concise information, we will use these ordinal notations until the end of the proof. In such a framework, each element in $\mathcal{N}$ corresponds to a natural number and by Lemma 6, $f(0,\ldots,0,t)$ corresponds to $t+1$ if $t \notin \mathcal{N}$. Each $t_i$ may be written $\epsilon_i x_i + N_i$, where $\epsilon_i$ is 0 or 1 and $N_i$ is a natural number. This is so because $t_i$ is supposed to belong to $\mathcal{N}(X)$.

$t_k$ must be a variable (otherwise, there would be a variable $y$ occurring in $t_k$ such that $y\sigma \notin \mathcal{N}$ which contradicts the fact that all $t_j\sigma$, $j > k$ are in $\mathcal{N}$). Let $\alpha$ be the ordinal $t_k\sigma$. Now let $X_0$ be the set of variables $y$ such that $y\sigma = \alpha + N_y$ with $N_y < \omega$. We are going to show that the substitution $\theta$ defined by

- $x\theta \equiv t_{k+1}\sigma + N_x + 1$ if $x \in X_0$,
- $x\theta \equiv x\sigma$ otherwise,

is again a solution of $\mathcal{I}$ s.t. $x\theta < x\sigma$. We investigate then some cases showing that $\theta$ is a solution of $\mathcal{I}$

- If $t_m > t_j$ is in $\mathcal{I}$ and $m > k$, then obviously, $\theta$ is a solution of $t_m > t_j$

- If $t_m > t_j$ is in $\mathcal{I}$ and $m \leq k, j > k$, then $t_m\theta >_{lpo} t_{k+1}\sigma$ by definition of $\theta$. Therefore, $\theta$ is a solution of $t_m > t_j$

- If $t_m > t_j$ is in $\mathcal{I}$ and $t_j\sigma \geq \alpha + \omega$ then $x_j \notin X_0$. Thus $t_j\sigma \equiv t_j\theta$. In the same way, $t_m\sigma \equiv t_m\theta$. $\theta$ is again a solution.

- If $t_m > t_j$ is in $\mathcal{I}$ and $t_m\sigma \geq \alpha + \omega$, $t_j\sigma < \alpha + \omega$, then $t_m\theta \equiv t_m\sigma > t_j\sigma \geq t_j\theta$ and $\theta$ is a solution of $t_m > t_j$.

- Suppose now that $t_m > t_j$ is in $\mathcal{I}$ and that $\alpha + \omega > t_m\sigma > t_j\sigma \geq \alpha$. This means that $x_m, x_j \in X_0$: $x_m\sigma = \alpha + N_m > x_j\sigma = \alpha + N_j$. Then

$$x_m\theta = t_{k+1}\sigma + 1 + N_m > x_j\theta = t_{k+1}\sigma + 1 + N_j.$$

  $\theta$ is again a solution of $t_m > t_j$.

This proves that $\theta$ is a solution of $\mathcal{I}$: all possible cases for $t_m > t_j$ have been investigated. On the other hand, $\theta < \sigma$ which contradicts the minimality hypothesis on $\sigma$. □

**Corollary 1** . *The satisfiability of natural simple systems is decidable.*

This is an easy consequence of Lemma 8 because we get linear inequations over the integers.

### *4.3. From Simple Systems to Natural Simple Systems*

The rules given in Figs. 4 and 5 transform any simple system into either $\top$, $\perp$ or a finite disjunction of simple systems. They preserve the existence of a solution. More precisely, we shall say that a rule $(S_i)$ is *correct* if, for every simple system $\mathcal{I}$, $\mathcal{I} \rightsquigarrow_{S_i} \bigvee_{j \in J} c_j$ implies that each $c_j$ is a simple system and $\mathcal{I}$ has a solution iff one of the $c_j$'s has a solution. In particular, when $J = \emptyset$, i.e. $\mathcal{I} \rightsquigarrow \perp$, $\mathcal{I}$ has no solution.

In the definition of the rules of Figs. 4 and 5 we adopt the following conventions:

- $\mathcal{I} \equiv t_1 > \ldots > t_n$ is the simple system on which the rule is applied

- If $\mathcal{I} \equiv t_1 > \ldots > t_n$ where $t_1, \ldots, t_n$ are ground terms (we call it a *ground system*), then $\mathcal{I} \Rightarrow^*_{\mathcal{R}} \top$. In such a case, $\mathcal{I}$ is obviously satisfiable. Therefore, we always assume that there is at least one term $t_i \notin T(F)$.

- $x_1$ is the greatest variable of $\mathcal{I}$ (i.e. the one occurring with the smallest index in $t_1, \ldots, t_n$. Such a variable exists since, by definition of simple systems, any subterm of a term $t_i$ is itself some $t_j$).

- The symbols $\gamma_i$ stand for any (possibly empty) sequence $t_k > \ldots > t_l$.

- A simple system consisting in an empty sequence of terms or in a single term must be understood as $\top$ .

**0 must be the rightmost term in the system**

$$(S_1) \quad \gamma_1 > 0 > \gamma_2 \quad \rightsquigarrow \quad \bot$$

if $\gamma_2$ is not empty.

$$(S_2) \quad \gamma > y \quad \rightsquigarrow \quad \gamma\{y \mapsto 0\} > 0 \vee \gamma > y > 0$$

if $y$ is a variable and 0 does not occur in $\gamma$.

**Elimination of the leftmost variable**

$$(S_3) \quad x_1 > \gamma_1 \quad \rightsquigarrow \quad \gamma_1$$

$$(S_4) \quad \gamma_1 > t[x_1] > \gamma_2 > x_1 > \gamma_3 \quad \rightsquigarrow \quad \gamma_1 > \gamma_2 > x_1 > \gamma_3$$

if $x_1$ does not occur in any term of the sequence $\gamma_1$.

**Elimination of the leftmost variable : the case $t \notin \mathcal{N}(X)$**

$$(S_5) \quad \gamma_1 > t > x_1 > \gamma_2 > 0 \quad \rightsquigarrow \quad \gamma_1 > t > \gamma_2 > 0$$

If $t \notin ST_\forall(X)$ and $x_1$ occurs only once in the problem.

$$(S_6) \quad \gamma_1 > t > x_1 > \gamma_2 \quad \rightsquigarrow \quad \bot$$

If $t \in ST_\forall(X) - \mathcal{N}(X)$ and $x_1$ occurs only once in the problem.

**Reduction to natural simple systems when $t \in \mathcal{N}(X)$**

$$(S_7) \quad \gamma_1 > c > x_1 > \gamma_2 \quad \rightsquigarrow \quad c > x_1 > \gamma_2$$

If $c \in C$ and $\gamma_1$ is not empty.

$$(S_8) \quad \gamma_1 > t > x_1 > \gamma_2 > t' > \gamma_3 \quad \rightsquigarrow \quad \bot$$

If $t \in \mathcal{N}(X)$, $t' \notin \mathcal{N}(X)$ and $x_1 \notin Var(\gamma_1, t)$.

Fig. 4. Transformation rules for simple systems

$$(S_9) \quad \gamma_1 > f(0,\ldots,0,t) > x_1 > \gamma_2 > 0 \quad \rightsquigarrow \quad f(0,\ldots,0,t) > x_1 > \gamma_2 > 0$$

if

- Every term in the sequence $\gamma_2$ belongs to $\mathcal{N}(X)$,
- $\gamma_1$ is not empty,
- $x_1$ occurs only once in the problem,
- $t \in \mathcal{N}(X)$.

Fig. 5. Transformation rules for simple systems (continued)

The rules $S_1, S_2$ force the last term $t_n$ to be 0. The rule $S_3$ removes the largest variable $x_1$ if there is no upper bound constraint on it. $S_4$ removes all occurrences of $x_1$ but one. $S_5$, $S_6$ investigate the cases where the term $t$ just before $x_1$ is not in $\mathcal{N}(X)$ : either the problem has no solution because $x_1$ must be between a term and its predecessor ($S_6$) or $x_1$ can be removed because $t$ cannot be a successor term ($S_5$). Finally, the rules $S_7, S_8, S_9$ assume $t \in \mathcal{N}(X)$ and remove from the system all terms that do not belong to $\mathcal{N}(X)$.

The following lemmas show the correctness of all rules ($S_i$).

**Lemma 9** . *($S_1$) is correct.*

**Proof.** This is a consequence of the definition of 0. □

**Lemma 10** . *($S_2$) is correct.*

**Proof.** $\gamma > y > 0$ is a simple system: every subterm of some term $t_i$ is some term $t_j$ because this property holds for $\gamma > y$. On the other hand, if

$$\gamma > y > 0 \Rightarrow_{\mathcal{R}} \bigvee_{j \in J} c_j,$$

either the rule which is applied does not involve 0 and, for some $j$, $c_j \equiv c'_j > 0$ where $c'_j$ is a subformula of $\gamma > y$ or else, the rule involves 0. In the latter case, $J$ contains one element (call it 1) and $c_1 \equiv \gamma > y$, because 0 is the least term.

$\gamma\{y \mapsto 0\} > 0$ is a simple system: if $s[t]$ is a term occurring in $\gamma\{y \mapsto 0\}$, then, replacing all occurrences of 0 with $y$, the corresponding term $s'[t']$ occurs in $\gamma$. $\gamma > y$ is a simple system. Therefore, $t'$ is one of the $t_i$'s. Then $t \equiv t'\{y \mapsto 0\}$ is a member of the sequence $\gamma\{y \mapsto 0\} > 0$. In the same way, if $\gamma\{y \mapsto 0\} > 0 \Rightarrow_R \bigvee_{j \in J} c_j$, then, either the transformation involves 0 or it can be lifted to a transformation $\gamma > y \Rightarrow_R \bigvee_{j \in J} c'_j$ where $c_j \equiv c'_j\{y \mapsto 0\}$ for every $j$. In both cases, there is a $c_j$ which is a subformula of $\gamma\{y \mapsto 0\} > 0$.

Finally, $\{y \mapsto t\}\sigma$ is a solution of $\gamma > y$ iff either $t \equiv 0$ and $\sigma$ is a solution of $\gamma\{y \mapsto 0\} > 0$ or else $t > 0$ and $\{y \mapsto t\}\sigma$ is a solution of $\gamma > y > 0$. □

**Lemma 11** . *$(S_3)$ is correct.*

**Proof.** $\gamma_1$ is obviously a simple system because $x_1$ does not occur in it. On the other hand, if $\sigma$ is a solution of the left hand side, it is obviously a solution of the right-hand side.

Now, if $\sigma$ is a solution of the right right-hand side, let $t_2$ be the leftmost term in $\gamma$ and let $u \equiv succ(t_2\sigma)$. Then $\theta = \{x_1 \mapsto u\}\sigma$ is a solution of the left-hand side. □

**Lemma 12** . *$(S_4)$ is correct.*

**Proof.** First, the right-hand side is a simple system because

- it is a subformula of the left-hand side,
- since $x_1$ does not occur in $\gamma_1$, there is no term in $\gamma_1$ having $t[x_1]$ as a subterm.

Now, we have to prove that any solution of the right-hand side is also a solution of the left-hand side. Assume that $\mathcal{I} \rightsquigarrow_{S_4} \mathcal{I}'$. Let $\bigvee_{j \in J} c_j$ be a solved form of $t_i > t[x_1]$ (resp. $t[x_1] > t_i$) and $c_{j_0}$ a subsystem of $\mathcal{I}$. Every inequation in $c_{j_0}$ has one of the following forms:

- $x_i > u$ : in this case $x_i > u$ is a subformula of $\mathcal{I}'$ since $x_1$ is the greatest variable
- $u > x_i$ and $u$ is different from $t[x_1]$. In this case $u > x_i$ is again a subformula of $\mathcal{I}'$
- $t[x_1] > x_i$ with $i \neq 1$. In this case, any solution of $\mathcal{I}'$ is also a solution of $x_1 > x_i$ and therefore a solution of $t[x_1] > x_i$.

In all cases, every solution of $\mathcal{I}'$ is also a solution of $c_{j_0}$. Therefore, any solution of $\mathcal{I}'$ is a solution of $\mathcal{I}$. The converse inclusion is obvious, hence the rule $(S_4)$ is correct. □

**Lemma 13** . *$(S_5)$ is correct.*

**Proof.** As in the previous lemma, the right-hand side is a simple system because it is a subformula of the left-hand side and the term which is erased $(x_1)$ does not occur elsewhere in the right-hand side.

Let $\mathcal{I} \rightsquigarrow_{S_5} \mathcal{I}'$. It is sufficient to prove that, if $\mathcal{I}'$ has a solution $\sigma$, then $\mathcal{I}$ has a solution $\theta$.

First, if $x \in Var(t)$, then $x\sigma >_{lpo} 0$ because $x > 0$ is a subformula of $\gamma_2$. Then, from lemmas 6 and 7, $t\sigma \notin ST$. (Actually, $t \in ST_\exists(X)$ implies that either $t \in C$ or $t \equiv f(y_1, \ldots, y_k, u)$ with at least one index $i$ such that $y_i \in X$. $t \notin \mathcal{N}$ and $y_i\sigma >_{lpo} 0$ then imply that $t\sigma \notin ST$).

Now let $\gamma_2 > 0 \equiv u > \gamma_3$ and $\theta = \{x_1 \mapsto succ(u\sigma)\}\sigma$. $\theta$ is a solution of $\mathcal{I}$: $x_1\theta >_{lpo} u\theta$ by construction, $t\theta \equiv t\sigma >_{lpo} x_1\theta$ because $t\sigma \notin ST$ and all other inequations are satisfied since they do not contain $x_1$. □

**Lemma 14** . *$(S_6)$ is correct.*

**Proof.** We only have to prove that the left-hand side has no solution. By Lemma 7 and because of the conditions imposed on $(S_6)$, $t \equiv f(0, \ldots, 0, u)$ with $u \notin \mathcal{N}(X)$. Then, by definition of $succ$, for every ground substitution $\sigma$, $t\sigma \equiv succ(u\sigma)$. On the other hand, $u$ is not a variable and it must occur in $\gamma_2$. A solution $\sigma$ of the left-hand side should therefore satisfy $succ(u\sigma) >_{lpo} x_1\sigma >_{lpo} u\sigma$ which is impossible, by Lemma 5. $\square$

**Lemma 15** . *$(S_7)$ is correct.*

**Proof.** $c > x_1 > \gamma_2$ is a simple system because

- it is a subformula of the left-hand side,
- $Sub(c > x_1 > \gamma_2) \cap \gamma_1 = \emptyset$. (Indeed, a term cannot occur twice in a sequence and a term cannot occur after one of its subterms).

We only have to prove that every solution $\sigma$ of the right-hand side is a solution of the left-hand side. For, let $s > t$ be a subformula of $\gamma_1 > c$. Let $\bigvee_{j \in J} c_j$ be a solved form of $s > t$ and $c_{j_0}$ be a subformula of $\gamma_1 > c > x_1 > \gamma_2$. Every inequation in $c_{j_0}$ has one of the following forms:

- $x_i > u$. In this case, the inequation occurs in $c > x_1 > \gamma_2$ because $x_1$ is the leftmost variable. Therefore, $x_i\sigma >_{lpo} u\sigma$.
- $u > x_i$ and $u$ is not a variable. Let $u \equiv g(u_1, \ldots, u_m)$. Either $u$ occurs in $c > x_1 > \gamma_2$ and $\sigma$ is a solution of $u > x_i$ or $u$ occurs in $\gamma_1$. In this latter case, $g >_F c$ (otherwise, $u > c \Rightarrow^*_{\mathcal{R}} \perp$). Thus, for every $\sigma$, $u\sigma >_{lpo} c$. In particular, since $\sigma$ is a solution of $c > x_i$, $u\sigma >_{lpo} x_i\sigma$.

We have proved that, in any case, $\sigma$ is a solution of $s > t$. Therefore, $\sigma$ is a solution of $\gamma_1 > c > x_1 > \gamma_2$. $\square$

**Lemma 16** . *$(S_8)$ is correct.*

**Proof.** First, if $t \in C$, the correctness is obvious since no constant in $C$ is greater than any instance of $t' \in \mathcal{N}(X)$. Assume now that $t \equiv f(0, \ldots, 0, v)$. Suppose that $\sigma$ is a solution of the left-hand side $\mathcal{I}$. Then $x_1\sigma >_{lpo} t'\sigma \notin \mathcal{N}$. By Lemma 6, a term in $\mathcal{N}$ cannot be larger than a term in $T(F) - \mathcal{N}$. Therefore $x_1\sigma \notin \mathcal{N}$. In the same way, $f(0, \ldots, 0, v)\sigma \notin \mathcal{N}$ and therefore $v\sigma \notin \mathcal{N}$. From Lemma 6, $f(0, \ldots, 0, v\sigma) = succ(v\sigma)$. But, as $v \not\equiv x_1$, there is an inequation $x_1 > v$ in $\mathcal{I}$. This means that $succ(v\sigma) > x_1\sigma > v\sigma$ which contradicts Lemma 5. $\square$

The crux of the proof is the correctness of $(S_9)$. We will use Lemma 8 and the following result:

**Lemma 17** . *In any simple system $t_1 > \ldots > t_n$, if $t_i \equiv g(u_1, \ldots, u_p)$ and $t_{i+1} \equiv h(v_1, \ldots, v_q)$, then $g \geq_F h$ or $t_{i+1}$ is a subterm of $t_i$.*

**Proof.** Suppose $h >_F g$, then

$$g(u_1, \ldots, u_p) > h(v_1, \ldots, v_q) \Rightarrow_{\mathcal{R}} \bigvee_{i=1,\ldots,p} u_i \geq h(v_1, \ldots, v_q)$$

and, by definition of simple systems, there is an index $i$ such that, either $u_i > h(v_1, \ldots, v_q)$ occurs in $\mathcal{I}$ or $u_i \equiv h(v_1, \ldots, v_q)$. On the other hand, $u_i$ has to occur somewhere in $\mathcal{I}$, and this must be after $g(u_1, \ldots, u_p)$: either $h(v_1, \ldots, v_q) > u_i$ occurs in $\mathcal{I}$ or $h(v_1, \ldots, v_q) \equiv u_i$. There is only one consistent combination of these two observations: $u_i \equiv h(v_1, \ldots, v_q)$. $\square$

**Lemma 18** . *$(S_9)$ is correct.*

**Proof.** By the control, every term in $\gamma_2$ belongs to $\mathcal{N}(X)$. (In particular, $t$ belongs to this set). Assume $\mathcal{I} \rightsquigarrow_{S_9} \mathcal{I}'$. By Lemma 8, $\mathcal{I}'$ has a solution iff it has a solution $\sigma$ such that every variable $x$ satisfies $x\sigma \in \mathcal{N}$. We are going to show that $\sigma$ is a solution of $\mathcal{I}$. Let $\gamma_1 > f(0, \ldots, 0, t) \equiv t_1 > \ldots > t_k$. (For convenience, we include here the case where $\gamma_1$ is empty.) We prove this result by induction on $k$. If $k = 1$, this is straightforward. Assume now $k \geq 2$.

Let $t_i > t_j$ be such that $i, j \leq k$ and let $\bigvee_{j \in J} c_j$ be one of its solved forms. Some problem $c_{j_0}$ is a subformula of $\mathcal{I}$. Let $u > v$ be an inequation in $c_{j_0}$. If $u \not\equiv t_i$, then $\sigma$ is a solution of $u > v$ because this inequation must either occur in $\mathcal{I}'$ or be an inequation $t_{i'} > t_{j'}$ with $i' > i$. In the latter case, we just use the induction hypothesis.

Assume now that $u \equiv t_i$ and $v$ is a variable. If $u \notin \mathcal{N}(X)$, then, from Lemma 6, $u\sigma >_{lpo} v\sigma$. Actually, $u \notin \mathcal{N}(X)$ is the only possible case. For, assume $u \in \mathcal{N}(X)$. From Lemma 17, $t_j$ is either a subterm of $t_i$ (and in this case, the solved form of $t_i > t_j$ would be $\top$) or $t_j \equiv f(u_1, \ldots, u_n)$. In the latter case,

$$\begin{aligned} f(0, \ldots, 0, v) > f(u_1, \ldots, u_n) \Rightarrow\ & v > f(u_1, \ldots, u_n) \\ & \vee (u_1 = 0 \wedge \ldots \wedge u_{n-1} = 0 \wedge v > u_n) \vee c_1 \vee \ldots \vee c_m. \end{aligned}$$

But, in each formula on the right, $t_i$ has been decomposed : it is not possible to find it again in any solved form. This contradicts the assumption $u \equiv t_i$. This shows that $\sigma$ is a solution of $t_i > t_j$ when $i, j \leq k$.

Assume now that $i \leq k - 1$ and $j > k$. Then $t_i\sigma >_{lpo} t_k\sigma$ and $t_k\sigma >_{lpo} t_j\sigma$, which proves again $t_i\sigma >_{lpo} t_j\sigma$. Thus, in any case, $\sigma$ is a solution of $\mathcal{I}$. $\square$

**Lemma 19** . *The system of rules given in figures 4 and 5 terminates. Moreover, any irreducible simple system is either a natural simple system or a ground system.*

**Proof.** The termination is straightforward : the number of terms in a simple system strictly decreases by application of any rule but $S_2$. On the other hand, $S_2$ may be applied at most once.

We have only to show that the rules cover all possible cases. Assume that $\mathcal{I} \equiv t_1 > \ldots > t_n$ contains a variable and a term $t \notin \mathcal{N}(X)$. For each rule $S_i$, assuming $\mathcal{I}$ irreducible w.r.t. $S_j$, $j < i$, we show below which additional properties $\mathcal{I}$ has if it is moreover assumed to be irreducible w.r.t. $S_i$.

**$S_1, S_2$:** $\mathcal{I} \equiv \gamma_1 > 0$

**$S_3, S_4$:** $\mathcal{I} \equiv \gamma_1 > t > x_1 > \gamma_2 > 0$ and $x_1$ occurs only once.

**$S_5$:** $t \in ST_\forall(X)$

**$S_6$:** $t \in \mathcal{N}(X)$

**$S_7$:** $t \in \mathcal{N}(X) - C$ i.e. $t \equiv f(0,\ldots,0,u)$ with $u \in \mathcal{N}(X)$ or else, $t \in C$ and $\gamma_1$ is empty.

**$S_8$:** Every term in $\gamma_2$ belongs to $\mathcal{N}(X)$

**$S_9$:** Now, all conditions for applying $S_9$ are fulfilled: there is a contradiction. □

Bringing all results of this section together, it is now possible to state:

**Theorem 1** . *The existence of a solution to a simple system is decidable.*

Together with the results of sections 2 and 3 we get:

**Theorem 2** . *The existence of a solution to an inequational problem is decidable.*

## 5. Further Remarks

- Our technique for deciding the existence of a solution is actually constructive: it is possible to extract an actual solution from the satisfiability proof. However, our method does not provide a "compact" representation of the set of all solutions. More precisely, assuming that some variables of an inequational problem are existentially quantified, our method does not provide an equivalent quantifier-free formula. That is the reason why the technique we have presented here cannot be lifted in an obvious way for deciding the first order theory of a lexicographic path ordering.
- Our algorithm has a high complexity. This is not surprising since the problem can actually be shown to be NP-hard.
- Theorem 2 and the technique we presented in this paper can be extended for solving inequations over arbitrary ordinal notations (J.-P. Jouannaud and M. Okada, private communication).
- The decidability of the first order theory of the lexicographic path ordering with a total precedence is still an open question. However, it has been shown by R. Treinen that, as soon as two function symbols are uncomparable w.r.t. $\geq_F$, then the first order theory of $\geq_{lpo}$ is undecidable[13].
- In this latter case, the decidability of the existential fragment of the theory is still open. (Our technique cannot be generalized in a straightforward way.)
- As mentioned in the introduction, Theorem 2 can be used for defining an extension of $>_{lpo}$ to non-ground terms which is more powerful than the usual extension[7] where variables are considered as new (unrelated) function symbols. Let $s >_{lpo*} t$ iff for every ground substitution $\sigma$, $s\sigma >_{lpo} t\sigma$. $>_{lpo*}$ is

decidable because the unsatisfiability of $s \leq t$ is decidable (theorem 2). $>_{lpo*}$ is a simplification ordering. Of course, it contains the usual extension $>_{lpo}$. This inclusion may be strict, as shown by the two examples;

**Example 9**. $f >_F 1 >_F 0$. $f(x, 1)$ and $f(0, 0)$ are uncomparable w.r.t. $>_{lpo}$. However, $f(x, 1) >^*_{lpo} f(0, 0)$ because any ground term is greater or equal to 0.

**Example 10**. $h >_F g >_F s >_F 0$. $u \equiv g(h(s(x), x), h(y, y))$ and $t \equiv h(s(x), y)$ are uncomparable w.r.t. $>_{lpo}$. On the other hand, $u >_{lpo*} t$. Indeed, $u \leq t \Rightarrow^*_{\mathcal{R}} y > x \wedge s(x) > y$ which has no solution.

**Acknowledgments**

I acknowledge J.-P. Jouannaud, M. Rusinowitch and R. Treinen for comments and discussions on an early version of this paper. I also thank an anonymous referee for his careful reading of the paper and a number of relevant comments.

## References

1. H. Comon and P. Lescanne, "Equational problems and disunification", *J. Symbolic Computat.* **7** (1989) 371-425.
2. M. J. Maher, "Complete axiomatizations of the algebras of finite, rational and infinite trees", in *Proc. 3rd IEEE Symp. Logic in Computer Science, Edinburgh*, July 1988, pp 348-357.
3. A. I. Mal'cev, "Axiomatizable classes of locally free algebras of various types", in *The Metamathematics of Algebraic Systems. Collected Papers, 1936-1967* (North-Holland, 1971) pp. 262-289.
4. H. Comon, "Disunification: a survey", in *Computational Logic: Essays in Honor of Alan Robinson*, eds J.-L. Lassez and G. Plötkin (MIT Press, 1991) to appear.
5. K. N. Venkataraman, "Decidability of the purely existential fragment of the theory of term algebras", *JACM*, **34**, 2 (1987) 492-510.
6. J. Hsiang and M. Rusinowitch, "On word problems in equational theories", in *Proc. 14th ICALP, Karlsruhe, LNCS 267* (Springer-Verlag, July 1987).
7. N. Dershowitz, "Termination of rewriting", *J. Symbolic Comput.*, **3**, 1 (1987), 69-115.
8. L. Bachmair, N. Dershowitz and J. Hsiang, "Orderings for equational proofs", in *Proc. 1st IEEE Symp. Logic in Computer Science, Cambridge, Mass.*, June 1986, pp. 346-357.
9. N. Dershowitz and J.-P. Jouannaud, "Rewrite Systems", in *Handbook of Theoretical Computer Science*, volume B, J. van Leeuwen ed. (North-Holland, 1990).

10. M. Rusinowitch, "Bounded deduction and application to completion procedures", unpublished draft (1989).

11. N. Dershowitz and Z. Manna, "Proving termination with multiset orderings", *Commun. ACM* **22**, 8 (1979) 465-476.

12. N. Dershowitz and M. Okada, "Proof-theoretic techniques for term rewriting", in *Proc. 3rd IEEE Symp. Logic in Computer Science, Edinburgh*, June 1988.

13. R. Treinen, "A new method for undecidability proofs of first order theories", Tech. Report A-09/90, Universität des Saarladandes, Saarbrücken, May 1990.