# On a monotonic semantic path ordering*

Alfons GESER
Universität Ulm, Abteilung KI, Oberer Eselsberg, 7900 Ulm, FRG
Phone: +49 731 502 4118, Email: geser@informatik.uni-ulm.de

May 14, 1993

**Abstract**

The semantic path ordering $>_{spo}$ is an ordering that allows to prove termination of term rewriting systems. Unlike other such orderings, it is not monotonic. We construct two monotonic suborderings $>_{cspo}$, $>_{mspo}$, of $>_{spo}$. Both orderings rely on reasonable assumptions on the underlying semantic ordering, and mirror Kamin/Lévy's termination proof method. Moreover, $>_{mspo}$ is shown to cover $>_{spo}$ up to the subterm property. In the case of the semantic ordering being a simplification quasiordering, the three orderings even coincide. Thus the Knuth/Bendix ordering turns out to be a special case of the semantic path ordering.

## 1 Introduction

Termination of term rewriting systems has a number of important applications since for a finite, terminating term rewriting system,

- every term can safely be rewritten into a normal form
- confluence is decidable by computation of critical pairs
- if confluence holds, the simple word problem is solvable
- the reachability problem (given $t, t'$, does $t \xrightarrow[R]{*} t'$ hold?) is decidable
- inductive proofs can be done using the transitive closure $\xrightarrow[R]{+}$ of the rewrite relation as inductive ordering
- and even more.

As is known, a rewrite system terminates if and only if, there is a termination ordering, i.e. a stable, monotonic, transitive, and terminating binary relation, that covers each rule of the rewrite system. Termination of rewrite systems is undecidable [**?**]. So the challenge is to design termination orderings more and more powerful. Most of the known termination orderings are simplification orderings,

*A version of this paper, without the last section, has been submitted to RTA-93

i.e. they contain the subterm relation. Known simplification orderings include the recursive path and decomposition orderings [**?**], [**?**], [**?**], [**?**], the monotonic interpretation orderings [**?**], [**?**], and the Knuth/Bendix ordering [**?**], [**?**].

On the one hand, simplification orderings are easy to handle. On the other hand, they contain the embedding relation, a fact that makes a termination proof impossible for a self-embedding rewrite system. Termination orderings which are not simplification orderings have been developed lately. For instance, the transformation orderings [**?**], [**?**], or the well rewrite orderings [**?**]. These orderings and the above mentioned simplification orderings have been implemented in various tools.

Despite its convincing power, the semantic path ordering $>_{spo}$ [**?**], [**?**] is not availabe in any tool. This is probably for the reason that $>_{spo}$ is no termination ordering since it is not monotonic. Hence the standard proof method for termination of rewriting does not apply. The particular proof method, Kamin and Lévy designed for this case, requires a more sophisticated technical apparatus.

This paper aims at showing how $>_{spo}$ can be altered to a *termination ordering* that mirrors Kamin/Lévy's proof method. First we briefly review their approach, where we put stress on stability and monotonicity conditions. Then we develop the termination ordering $\gtrsim_{cspo}$ based on a semantic congruence. If the semantic quasiordering $\gtrsim\!\!\!\!\approx$ even is monotonic, then there is a stronger termination ordering $\gtrsim_{mspo}$. This ordering satisfies the nice property that it covers $\gtrsim_{spo}$ up to the subterm property. More precisely, $\gtrsim_{spo} = (\gtrsim_{mspo} \cup \rhd)^*$ holds. If the semantic ordering $\gtrsim\!\!\!\!\approx$ satisfies the subterm property, then $\gtrsim_{spo}$ is a simplification quasiordering, and moreover, $\gtrsim_{spo}$, $\gtrsim_{cspo}$, and $\gtrsim_{mspo}$ coincide. As a consequence, the Knuth/Bendix ordering is a special case of the semantic path ordering.

## 2 Preliminaries

We assume that the reader is familiar with term rewriting. For surveys on term rewriting see [**?**], [**?**], [**?**], and [**?**].

Let two disjoint sets $\mathcal{F}$ of *function symbols* and $\mathcal{X}$ of *variables* be preassumed together with a function arity : $\mathcal{F} \to \mathbb{N}$ which assigns each function symbol its fixed number of arguments. The set $\mathcal{T}$ of terms upon $\mathcal{F}$ and $\mathcal{X}$ is defined to be the smallest set containing $\mathcal{X}$ and satisfying

$$\text{arity}(f) = n \quad \text{and} \quad t_1, \ldots, t_n \in \mathcal{T} \qquad \text{implies} \qquad (f, t_1, \ldots, t_n) \in \mathcal{T}$$

Function symbols $f$ may also be seen as term constructing functions $f : \mathcal{T}^n \to \mathcal{T}$ by $f(t_1, \ldots, t_n) = (f, t_1, \ldots, t_n)$, a fact that allows to scrap the tuple notation.

A *substitution* $\sigma$ is an endomorphism on $\mathcal{T}$, i.e. a function $\sigma : \mathcal{T} \to \mathcal{T}$ that satisfies $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$. Application of a substitution will be denoted by postfixing the substitution. Because it is a homomorphism, $\sigma$ is uniquely given by its restriction to the mapping $\sigma : \mathcal{X} \to \mathcal{T}$. The set of all substitutions on $\mathcal{T}$ will be denoted by $\mathcal{S}ub$.

A transitive, irreflexive relation is called a strictordering, or an *ordering*, if no confusion arises. An equivalence relation $\sim$ is called an *associated equivalence* to an ordering $>$, if it is *absorbed* by $>$, i.e. if both $\sim > \subseteq >$ and $> \sim \subseteq >$ hold.

A *quasiordering* is a transitive, reflexive relation on terms. The union $> \cup \sim$ of an ordering with an associated equivalence is a quasiordering. Conversely, every quasiordering $\gtrsim$ defines an ordering $> =_{\text{def}} \gtrsim \setminus \lesssim$ together with an associated equivalence $\sim =_{\text{def}} \gtrsim \cap \lesssim$ where $\lesssim$ is the inverse of $\gtrsim$, i.e. $s \lesssim t$ means $t \gtrsim s$.

The *lexicographic combination* $(\gtrsim_1, \gtrsim_2)$ of two quasiorderings $\gtrsim_1$ and $\gtrsim_2$, is the quasiordering

$$(\gtrsim_1, \gtrsim_2) \ =_{\text{def}} >_1 \cup (\sim_1 \cap \gtrsim_2)$$

Let $\rightarrow$ denote any binary relation on terms. A $\rightarrow$-derivation is a sequence of steps $t_1 \rightarrow t_2 \rightarrow \ldots$, that may be finite or infinite. The relation $\rightarrow$ is called *terminating*, if every $\rightarrow$-derivation is finite. The transitive closure of $\rightarrow$ will be denoted by $\xrightarrow{+}$, its reflexive closure by $\xrightarrow{\varepsilon}$, and its transitive, reflexive closure by $\xrightarrow{*}$. By abuse of notation, we call a quasiordering $\gtrsim$ terminating, if its strict part $>$ is terminating. It is known that the lexicographic combination of two terminating quasiorderings terminates.

$\rightarrow$ is called *monotonic*, if everywhere

$$t \rightarrow t' \qquad \text{implies} \qquad f(t_1, \ldots, t, \ldots, t_n) \rightarrow f(t_1, \ldots, t', \ldots, t_n)$$

and *stable*, if everywhere

$$t \rightarrow t' \qquad \text{implies} \qquad t\sigma \rightarrow t'\sigma$$

A rewrite system $R$ is any binary relation on terms. The *$R$-rewrite relation* $\underset{R}{\rightarrow}$ is defined as the stable and monotonic closure of $R$. It is wellknown that the rewrite step (given $R, t$, wanted $t'$ such that $t \underset{R}{\rightarrow} t'$ holds) is computable. $R$ is called a *terminating rewrite system*, if its rewrite relation $\underset{R}{\rightarrow}$ terminates. A *termination ordering* is a monotonic, stable, transitive, and terminating relation on terms (irreflexivity follows from termination). Do not confuse *termination* ordering with *terminating* ordering. The latter does not require any monotonicity and stability conditions. A termination ordering is the basis to prove termination of a rewrite system: $R$ is a terminating rewrite system, if and only if, for some termination ordering $>$, one has $R \subseteq >$, i.e. $l > r$ holds for all rules $l \rightarrow r \in R$. In particular, a terminating rewrite system $R$ defines a termination ordering $\xrightarrow[R]{+}$, because the transitive closure preserves termination. See Dershowitz [**?**] for a comprehensive survey on termination of rewriting.

Monotonic orderings and monotonic quasiorderings, remarkably, do *not* depend on each other. If a quasiordering $\gtrsim$ is monotonic, by symmetry its equivalence part $\sim$ is monotonic as well, i.e. we have

$$t \sim t' \qquad \text{implies} \qquad f(t_1, \ldots, t, \ldots, t_n) \sim f(t_1, \ldots, t', \ldots, t_n)$$

A monotonic equivalence is also called a *congruence*. The strict part of a monotonic quasiordering needs not be monotonic. Conversely, given a monotonic ordering $>$, one cannot be sure that some associated equivalence $\sim$ is monotonic, too, except for the trivial case where $\sim$ coincides with $=$. A monotonic ordering together with an associated congruence, however, define a monotonic quasiordering. The same reasoning applies, mutatis mutandis, also for stability instead of monotonicity.

Let $\gtrsim = (\gtrsim_1, \gtrsim_2)$ be the lexicographic combination of two quasiorderings $\gtrsim_1$ and $\gtrsim_2$. If both $\sim_1$ and $\sim_2$ are monotonic, then so is $\sim$. If moreover $\gtrsim_1$ is monotonic, then so is $\gtrsim$. If all $>_1, \sim_1, >_2, \sim_2$ are monotonic, then so are $>, \sim$. But note that $\gtrsim_1, \gtrsim_2$ monotonic does not suffice to ensure $\gtrsim$ monotonic.

A *termination quasiordering* is a stable and monotonic quasiordering whose strict part is a termination ordering. For a couple of reasons, we prefer termination quasiorderings:

- for the framework of $E$-termination [**?**], where stability and monotonicity of $\sim$ are useful in order to show $\leftrightarrow_E \subseteq \sim$
- lexicographic combination of termination quasiorderings yields again a termination quasiordering ([**?**] for polynomial interpretations). A corresponding property for orderings instead of quasiorderings does not hold.
- path orderings and their corresponding quasiorderings are defined mutually recursively. Then for technical reasons, both (the ordering and the quasiordering) ought to be monotonic and stable.
- termination quasiorderings allow to prove relative termination [**?**].

The *superterm* order $\rhd$ on terms is the stable and transitive closure of

$$\{f(x_1, \ldots, x_n) \rhd x_i \mid f \in \mathcal{F}, \quad \text{arity}(f) = n, \quad i \in \{1, \ldots, n\}\}$$

Intuitively, $t \rhd t'$ means that $t$ is of the form $c(t')$ for some nonempty context $c(.)$. The *subterm* order $\lhd$ is the inverse of $\rhd$. The quasiorderings $\trianglerighteq$ and $\trianglelefteq$ denote the reflexive closure of $\rhd$ and $\lhd$, respectively. A binary relation on terms $> \supseteq \rhd$ is also said to possess the *subterm property*. A monotonic quasiordering $\gtrsim$ extending $\rhd$ is called a *simplification quasiordering*. Each simplification quasiordering by definition extends the relation $\xrightarrow[\rhd]{*}$, the inverse $\xrightarrow[\lhd]{*}$ of which is known as the *embedding* relation. By Kruskal's tree theorem, such a quasiordering is terminating.

In contrast, a *simplification ordering* $>$ is a monotonic ordering satisfying the slightly stronger condition $\rhd \subseteq >$. Every simplification ordering defines a simplification quasiordering via reflexive closure. Conversely, a simplification quasiordering $\gtrsim$ defines a simplification ordering $>'$ by $\gtrsim' =_{\text{def}} (\gtrsim, \xrightarrow[\rhd]{*})$, provided the strict part $>$ is monotonic.

Given a quasiordering $\gtrsim$ on a set $S$, its *multiset extension* is a quasiordering $\gtrsim_{mult}$ on the set $S^*$ of $S$-sequences which is defined as follows:

$s \gtrsim_{mult} t$, if

- $t = ()$, or
- $t = bt', \quad a \in s, \quad a \sim b, \quad s - a \gtrsim_{mult} t - b$, or
- $t = bt', \quad \forall a' \sim b. a' \notin s, \quad a \in s, \quad a > b, \quad s >_{mult} t - b.$

Here sequence concatenation is denoted by juxtaposition, $a \in s$ means that $a$ occurs in $s$, and $s - a$ means $s$ where the first occurrence of $a$ is removed. Usually,

the multiset extension is defined on the set of multisets over $S$, but it appears more convenient for the purposes of this paper to define it on $S^*$.

The *lexicographic extension* of a quasiordering $\gtrsim$ on a set $S$ towards a quasiordering $\gtrsim_{lex}$ on $S^*$ is defined by:

$s \gtrsim_{lex} t$, if

- $t = ()$, or
- $t = bt', \quad s = as', \quad a > b$, or
- $t = bt', \quad s = as', \quad a \sim b, \quad s' \gtrsim_{lex} t'$.

# 3 Status Extension Operators

The semantic path ordering is a powerful generalization of the recursive path ordering. It uses a recursive scheme that has two parameters: A semantic quasiordering, and a status extension operator. As the status extension operator is essential for the understanding of the semantic path ordering, we must care about it beforehand.

A status extension operator $\_^{stat}$ is a mapping from a quasiordering $\gtrsim$ on terms to an ordering $\gtrsim_{stat}$ on terms which relies on $\gtrsim$-comparisons of subterms only.

**Definition 3.1 (Status extension, $\_^{stat}$, [?])** *Let $\mathcal{R}el(\mathcal{T})$ denote the set of binary relations on terms. A mapping $\_^{stat} : \mathcal{R}el(\mathcal{T}) \to \mathcal{R}el(\mathcal{T})$ that maps a stable quasiordering $\gtrsim$ where $>$ is stable, towards a relation $\gtrsim_{stat}$ on non-variable terms is called a* status extension *operator, if it satisfies the following conditions:*

- *$\gtrsim_{stat}$ is transitive.*
- *both $>_{stat}$ and $\sim_{stat}$ are stable.*
- *both $>_{stat}$ and $\sim_{stat}$ are* pseudo-monotonic*:*
  $t > t'$ *implies* $f(t_1, \ldots, t, \ldots, t_n) >_{stat} f(t_1, \ldots, t', \ldots, t_n)$,
  $t \sim t'$ *implies* $f(t_1, \ldots, t, \ldots, t_n) \sim_{stat} f(t_1, \ldots, t', \ldots, t_n)$.
- *the mapping $\_^{stat}$ is* subterm founded*, i.e. for all $s$, $t$ given, $s \gtrsim_{stat} t$ is equivalent to $s \gtrsim_{fin,stat} t$ where $\gtrsim_{fin,stat}$ denotes $\_^{stat}$ applied to the subrelation $\gtrsim_{fin}$ of $\gtrsim$ defined by*

$$\gtrsim_{fin} =_{\text{def}} \gtrsim \cap \{(s', t') \mid s' \trianglelefteq s, \quad t' \trianglelefteq t, \quad (s', t') \neq (s, t)\}$$

- *the mapping $\_^{stat}$* preserves termination*, i.e. for every infinite derivation $t_1 >_{stat} t_2 >_{stat} \cdots$, there is an infinite derivation $u_1 > u_2 > \cdots$ such that $\exists i \in \mathbb{N}. u_1 \triangleleft t_i$ holds.*

By abuse of notation, $>_{stat}$, $\sim_{stat}$ denote the strict part and the equivalence part of $\gtrsim_{stat}$, respectively. In contrast to Kamin/Lévy's definition, we introduce the operator as a mapping from quasiorderings to quasiorderings, rather than from orderings to orderings. This is for flexibility reasons: The status extension may also use $\sim$ calls for its work. As a consequence, we had to add pseudo-monotonicity

of $\sim$. From pseudo-monotonicity of $\sim$, reflexivity of $\gtrsim_{stat}$ follows. The stability conditions are needed because we work with terms containing variables. Termination preservation is necessary for the proof that the semantic path ordering is a terminating ordering. Our condition is very little looser than Kamin/Lévy's. Kamin and Lévy require a further condition, which we cover by our condition of subterm foundedness. It is essentially this:

- $\_{}_{stat}$ is $\subseteq$-continuous, i.e. $s \gtrsim_{stat} t$ is equivalent to $s \gtrsim_{fin,stat} t$ where $\gtrsim_{fin,stat}$ denotes $\_{}_{stat}$ applied to a finite subrelation $\gtrsim_{fin} =_{\text{def}} >_{fin} \cup \sim_{fin}$, $>_{fin} \subseteq >$, $\sim_{fin} \subseteq \sim$ of $\gtrsim$.

The continuity condition is needed for computational induction proofs of properties of the semantic path ordering. Subterm foundedness is a little bit stronger than continuity, for the status operator may no longer use $s' \gtrsim_{fin} t'$ for arbitrary $s', t'$, and the ordering $\gtrsim_{fin}$ is indeed finite because there are only finitely many subterms of a given term. But a subterm founded status extension enjoys two valuable advantages.

- Any terminating procedure for status extension turns the recursive definition of the semantic path ordering (see below) into a terminating procedure. Subterm foundedness cares for the proof that the procedure terminates.
- Proofs of properties of the semantic path ordering can be done without using computational induction. In fact, structural induction on tuples of terms is sufficient.

In practice, one uses a wellknown standard status extension operator. It is known that the lexicographic extension and the multiset extension can be used as status extensions. People use a status assignment $\text{stat} : \mathcal{F} \to \{mult, lr, rl\}$, telling that dependent on $f$, the sequence of arguments has to be compared as a multiset, or lexicographically, from left to right, or from right to left, respectively. More generally, any permutation $\pi$ of the argument positions may be used as lexicographic status.

**Definition 3.2 (Standard status extension, $\_{}_{stan}$)** *The* standard status extension *operator* $\_{}_{stan}$ *is defined by*

$f(s_1, \ldots, s_m) \gtrsim_{stan} g(t_1, \ldots, t_n)$, *if*

- $\text{stat}(f) = \text{stat}(g) = mult, \quad (s_1, \ldots, s_m) \gtrsim_{mult} (t_1, \ldots, t_n)$, *or*
- $\text{stat}(f) = \pi, \quad \text{stat}(g) = \rho, \quad (s_{\pi(1)}, \ldots, s_{\pi(m)}) \gtrsim_{lex} (t_{\rho(1)}, \ldots, t_{\rho(n)})$

If $f$, $g$ have incompatible status, then the terms are not comparable. It seems however safe to treat the cases where one of $f$, $g$ is unary as if it had arbitrary status.

As can easily be verified,

**Proposition 3.1** $\_{}_{stan}$ *is a status extension operator.*

# 4 The Semantic Path Ordering

In this section, we summarize what is known about the construction of $>_{spo}$. Having a notion of status extension, we can define the semantic path ordering.

**Definition 4.1 (Semantic path ordering, $\gtrsim_{spo}$, [?], [?])** *Let a status extension $\_{}_{stat}$ be given. Let moreover a stable quasiordering $\succsim$ be given where $\succ$ is stable and terminates. The semantic path quasiordering $\gtrsim_{spo} \subseteq \mathcal{T} \times \mathcal{T}$ is then defined as follows.*

$s \gtrsim_{spo} t$, *if*

- $s \trianglerighteq x = t$, *or*
- $s = f(s_1, \ldots, s_m), \quad t = g(t_1, \ldots, t_n)$, *and*
  - $s \succ t, \quad \forall i. s >_{spo} t_i$, *or*
  - $s \approx t, \quad s \gtrsim_{spo,stat} t, \quad \forall i. s >_{spo} t_i$, *or*
  - $\exists i. s_i \gtrsim_{spo} t$

It can be shown that the semantic path ordering satisfies all conditions for a termination ordering, except monotonicity.

**Theorem 1 ([?])** *1. $\gtrsim_{spo}$ is a stable quasiordering.*

*2. The semantic path ordering $>_{spo}$ is stable and terminating, and contains the superterm relation $\rhd$.*

**Proof ([?]; sketch)** *First one proves that $s \gtrsim_{spo} t$ implies that all variables of $t$ are also in $s$. From this, one can prove transitivity of $\gtrsim_{spo}$. By simultaneous induction, one can prove reflexivity and the subterm property. The stability properties follow from this. All these proofs are done by structural induction.*

*Termination is proven by "minimal counterexample". Assume $>_{spo}$ is not terminating. Then there is a minimal, infinite derivation $t^1 >_{spo} t^2 >_{spo} \ldots$. This derivation is minimal in the following sense: For all $i \in \mathbb{N}$, no infinite derivation that starts with $t^1 >_{spo} \ldots >_{spo} t^{i-1}$, continues with a term $t' \lhd t^i$. We are going to demonstrate by a case analysis on the structure of $>_{spo}$ that such a minimal counterexample cannot exist and so, that $>_{spo}$ terminates.*

*No step $t^i >_{spo} t^{i+1}$ in the given derivation can be due to the base case $t^i \rhd x = t^{i+1}$, since then the derivation would be finite, nor can it be due to the case $\exists j. t^i_j \gtrsim_{spo} t^{i+1}$, since then the infinite derivation*

$$t^1 >_{spo} \ldots >_{spo} t^{i-1} >_{spo} t^i_j >_{spo} t >_{spo} \ldots$$

*would be smaller, a contradiction to the minimality assumption. The case $t^i \succ t^{i+1}$, $\forall j. t^i >_{spo} t^{i+1}_j$ can occur only finitely many times, because $\succ$ terminates. So we may assume that from a certain point $N$ in the infinite derivation, only the case $t^i \approx t^{i+1}$, $t^i \gtrsim_{spo,stat} t^{i+1}$, $\forall j. t^i >_{spo} t^{i+1}_j$ is used. I.e. we have an infinite derivation $t^N >_{spo,stat} t^{N+1} >_{spo,stat} \ldots$. Termination preservation of the status*

*extension operator gives us the infinite derivation* $u^1 >_{spo} u^2 >_{spo} \dots$ *where for some* $k \geq N$, $u^1 \lhd t^k$ *holds. This is sufficient to construct the smaller derivation*

$$t^1 >_{spo} \dots >_{spo} t^{k-1} >_{spo} u^1 >_{spo} u^2 >_{spo} \dots$$

*a contradiction to the assumption that the given derivation was minimal.*

The fact that $>_{spo}$ is not monotonic, sets a problem. We must be aware of infinitely many rewrite steps $t \underset{R}{\longrightarrow} t'$, even if $R$ is finite. How do we prove $t >_{spo} t'$ for all these? $>_{spo}$ is stable. If moreover $>_{spo}$ were monotonic then we needed to prove only that $R \subset >_{spo}$. But then, $>_{spo}$ would be a simplification ordering, an unwelcome side effect. We are going to see in the next section how to dodge this problem.

# 5 A Monotonic Semantic Path Ordering

Although $>_{spo}$ is not monotonic, suborderings of $>_{spo}$ can be constructed which are monotonic. One such subordering, $>_{cspo}$, will be constructed in this section, another one, $>_{mspo}$ in a later section.

The key is the following property, introduced by Kamin and Lévy:

(1) $$t \underset{R}{\longrightarrow} t' \in R \quad \text{and} \quad t >_{spo} t' \qquad \text{implies}$$
$$f(x_1, \dots, t, \dots, x_n) \approx f(x_1, \dots, t', \dots, x_n)$$

**Lemma 5.1 ([?])** *From property (??), it follows that the relation* $\overset{+}{\underset{R}{\longrightarrow}} \cap >_{spo}$ *is monotonic.*

**Proof** *Let* $t \overset{+}{\underset{R}{\longrightarrow}} t'$ *and* $t >_{spo} t'$. *By monotonicity of* $\underset{R}{\longrightarrow}$,

$$f(x_1, \dots, t, \dots, x_n) \overset{+}{\underset{R}{\longrightarrow}} f(x_1, \dots, t', \dots, x_n)$$

*holds. So we have left to prove*

$$f(x_1, \dots, t, \dots, x_n) >_{spo} f(x_1, \dots, t', \dots, x_n)$$

*By definition of* $\gtrsim_{spo}$, *this amounts to*

1. $f(x_1, \dots, t, \dots, x_n) \approx f(x_1, \dots, t', \dots, x_n)$
2. $f(x_1, \dots, t, \dots, x_n) >_{spo,stat} f(x_1, \dots, t', \dots, x_n)$
3. $\forall i. f(x_1, \dots, t, \dots, x_n) >_{spo} x_i$
4. $f(x_1, \dots, t, \dots, x_n) >_{spo} t'$

*Claim (??) holds by definition of condition (??). Claim (??) holds by the pseudo-monotonicity of status extension. (??) and (??) are a consequence of the subterm property of* $>_{spo}$, *its transitivity, and the premise* $t >_{spo} t'$.

Thus $\xrightarrow[R]{+} \cap >_{spo}$ is a termination ordering, and so, in order to prove that $R$ is a terminating rewrite system, we just need to show that $R \subseteq \xrightarrow[R]{+} \cap >_{spo}$, which is the same as $R \subseteq >_{spo}$. This part is as comfortable as possible.

Condition (**??**) however, must still be shown for all, again potentially infinitely many, $t \underset{R}{\longrightarrow} t'$. Again one would prefer to prove the condition for $l \longrightarrow r \in R$ only.

$$(2) \qquad l \longrightarrow r \in R \quad \text{and} \quad l >_{spo} r \qquad \text{implies}$$
$$f(x_1, \ldots, l, \ldots, x_n) \approx f(x_1, \ldots, r, \ldots, x_n)$$

Fortunately, one may replace (**??**) by (**??**), by the observation that $\approx$ is a congruence.

**Proposition 5.1** *If $\approx$ is a congruence, then (??) and (??) are equivalent.*

So far, this is Kamin/Lévy's method to go without monotonicity of $>_{spo}$. The property that $\xrightarrow[R]{+} \cap >_{spo}$ be monotonic, is what Dershowitz [**?**] calls *monotonic for derivations*. This definition expresses the fact that the termination ordering $\xrightarrow[R]{+} \cap >_{spo}$ depends on a particular rewrite system $R$. This dependency is inconvenient. We prefer a termination ordering that needs not be master-tailored for each rewrite system, just as we are used to, with other termination orderings. Next we will show how to get one such. It is easy to see that the set

$$L =_{\text{def}} \{l \longrightarrow r | f(x_1, \ldots, l, \ldots, x_n) \approx f(x_1, \ldots, r, \ldots, x_n)\}$$

is the largest rewrite system $L$ that satisfies condition (**??**), in the sense that $\xrightarrow[L]{+}$ is maximal wrt. $\subseteq$. Therefore we define the greatest termination ordering $\xrightarrow[L]{+} \cap \gtrsim_{spo}$ below $\gtrsim_{spo}$, for which Kamin/Lévy's method applies, as follows.

**Definition 5.1 (Semantic quasiordering, $\succsim$)** *A* semantic quasiordering $\succsim$ *is a quasiordering such that $\approx$ is stable and monotonic, and $\succ$ is stable and terminating.*

**Definition 5.2 (Monotonic spo based on a congruence, $\gtrsim_{cspo}$)** *Let $\succsim$ be a semantic quasiordering. The restriction $\gtrsim_{cspo}$ ("congruent spo") of the semantic path quasiordering is defined by*

*$s \gtrsim_{cspo} t$, if*

- *$s \gtrsim_{spo} t$, and*
- *$\forall f \in \mathcal{F}, x_1, \ldots, x_n \in \mathcal{X}. \quad f(x_1, \ldots, s, \ldots, x_n) \approx f(x_1, \ldots, t, \ldots, x_n)$*

**Theorem 2** *1. $\sim_{cspo} = \sim_{spo}$.*

*2. $\gtrsim_{cspo}$ is a termination quasiordering.*

Now finally, we are completely back in a familiar framework. In order to show that $R$ is a terminating rewrite system, we provide a termination ordering $>_{cspo}$ for which we can show $R \subseteq >_{cspo}$. This amounts to nothing else than to show that $l >_{spo} r$ and (**??**) holds for all $l \longrightarrow r \in R$. Moreover we may completely dispose with the notion of monotonicity for derivations.

# 6 Suitable Semantic Orderings

In order to exercise examples, one must fix, besides a status extension operator, some semantic quasiordering $\succsim$. In the simplest case, $\succsim$ is defined by means of a precedence quasiordering.

**Definition 6.1 (Precedence, $\gtrsim$)** *A* precedence *is a terminating quasiordering $\gtrsim$ on $\mathcal{F}$.*

A precedence is lifted to a semantic quasiordering $\gtrsim_{\mathcal{F}}$ on terms in the following way.

**Definition 6.2 (Precedence quasiordering on terms, $\gtrsim_{\mathcal{F}}$)** *Let $\gtrsim$ be a precedence. The precedence quasiordering* on terms *$\gtrsim_{\mathcal{F}}$ is defined by*

$s \gtrsim_{\mathcal{F}} t$, *if*

- $s = x = t$, *or*
- $s = f(s_1, \ldots, s_m), \quad t = g(t_1, \ldots, t_n), \quad f \gtrsim g$

**Proposition 6.1** *$\gtrsim_{\mathcal{F}}$ is a semantic quasiordering.*

So one may set $\succsim = \gtrsim_{\mathcal{F}}$, a choice that turns the semantic path ordering into the recursive path ordering with status. Another possibility is to define $\succsim$ by an interpretation function.

**Definition 6.3 (Interpretation, $[\_]$)** *Let a set $\mathcal{D}$ be given, equipped with a terminating quasiordering $\gtrsim_{\mathcal{D}}$. An* interpretation *$[\_] : \mathcal{F} \to (\mathcal{D}^n \to \mathcal{D})$ is a mapping of $n$-ary function symbols into $n$-ary $\sim_{\mathcal{D}}$-monotonic functions on $\mathcal{D}$.*

That is to say, each interpretation $[f]$ of a function symbol $f \in \mathcal{F}$ satisfies

$$\forall d, d', d_1, \ldots, d_n \in \mathcal{D}.$$
$$d \sim_{\mathcal{D}} d' \qquad \text{implies} \qquad [f](d_1, \ldots, d, \ldots, d_n) \sim_{\mathcal{D}} [f](d_1, \ldots, d', \ldots, d_n)$$

The interpretation is extended in a straightforward way to a mapping $[\_] : \mathcal{T} \to (\mathcal{D}^n \to \mathcal{D})$ of terms containing variables $x_1, \ldots, x_n$ into $n$-ary $\sim_{\mathcal{D}}$-monotonic functions on $\mathcal{D}$. Now via $[\_]$, the quasiordering $\gtrsim_{\mathcal{D}}$ on objects is lifted towards a quasiordering $\gtrsim_{[\_]}$ on terms. One must take care to ensure that both $>_{[\_]}$ and $\sim_{[\_]}$ are stable.

**Definition 6.4 (Interpretation ordering, $\gtrsim_{[\_]}$)**

- $t >_{[\_]} t' \iff_{\text{def}} \forall d_1, \ldots, d_n.[t](d_1, \ldots, d_n) >_{\mathcal{D}} [t'](d_1, \ldots, d_n)$
- $t \sim_{[\_]} t' \iff_{\text{def}} \forall d_1, \ldots, d_n.[t](d_1, \ldots, d_n) \sim_{\mathcal{D}} [t'](d_1, \ldots, d_n)$
- $\gtrsim_{[\_]} =_{\text{def}} >_{[\_]} \cup \sim_{[\_]}$

By definition, $\sim_{[\_]}$ is a congruence. Moreover $>_{[\_]}$ is terminating because $>_{\mathcal{D}}$ is.

**Proposition 6.2** *$\gtrsim_{[\_]}$ is a semantic quasiordering.*

Apparently $\gtrsim_{[\_]}$ may be used for $\gtrapprox$, but as Kamin and Lévy suggest, one better combines a precedence with an interpretation lexicographically: $\gtrapprox = (\gtrsim_{\mathcal{F}}, \gtrsim_{[\_]})$.

**Proposition 6.3** *Lexicographic combination of two semantic quasiorderings yields a semantic quasiordering.*

Let us now consider an example in order to demonstrate the construction of the semantic quasiordering and the use of $>_{cspo}$.

**Example 1 (Factorial, [?])** *Let the predecessor function p, addition, multiplication, and the factorial function* fac *on* $\mathbb{N}$ *be specified by the following rewrite system* F*:*

$$\begin{aligned}
(1)\quad & p(s(x)) &&\to\ x\\
(2)\quad & x+0 &&\to\ x\\
(3)\quad & x+s(y) &&\to\ s(x+y)\\
(4)\quad & x*0 &&\to\ 0\\
(5)\quad & x*s(y) &&\to\ x*y+x\\
(6)\quad & \mathrm{fac}(0) &&\to\ s(0)\\
(7)\quad & \mathrm{fac}(s(x)) &&\to\ s(x)*\mathrm{fac}(p(s(x)))
\end{aligned}$$

*Any simplification quasiordering would fail to prove that the above rewrite system is a terminating rewrite system. The reason is that the rule* ***??*** *is self-embedding. In order to instantiate the semantic path ordering, one has to fix a status assignment and a semantic quasiordering. Any status assignment does the job. For the semantic quasiordering, we use* $\gtrapprox = (\gtrsim_{\mathcal{F}}, \gtrsim_{[\_]})$ *with* $\gtrsim_{\mathcal{F}}$ *defined by the precedence*

$$\mathrm{fac} > * > + > p > s > 0$$

*and* $\gtrsim_{[\_]}$ *defined by the intuitive interpretation*

$$\begin{aligned}
[0] &= 0\\
[s(x)] &= [x]+1\\
[p(x)] &= \begin{cases}[x]-1, & \text{if}\quad [x] >_{\mathbb{N}} 0\\ 0, & \text{else}\end{cases}\\
[x+y] &= [x]+[y]\\
[x*y] &= [x]*[y]\\
[\mathrm{fac}(x)] &= [x]!
\end{aligned}$$

*Here,* $>_{\mathbb{N}}$ *is to denote the natural ordering on* $\mathbb{N}$*. As it turns out,* $>_{cspo}$ *orders all rules of our example. Let us check this for the crucial rule (**??**).*

*The claim is* $\mathrm{fac}(s(x)) >_{cspo} s(x)*\mathrm{fac}(p(s(x)))$*: For the semantics, we have*

$$[\mathrm{fac}(s(x))] = ([x]+1)! = ([x]+1)*[x]! = [s(x)]*[p(s(x))]! = [s(x)*\mathrm{fac}(p(s(x)))]$$