Orderings for Innermost Termination

Mirtha-Lina Fernández[1] *, Guillem Godoy[2], and Albert Rubio[2]

[1] Universidad de Oriente, Santiago de Cuba, Cuba
Email: mirtha@csd.uo.edu.cu
[2] Universitat Politècnica de Catalunya, Barcelona, España
Email: ggodoy@lsi.upc.es,rubio@lsi.upc.es

Abstract. This paper shows that the suitable orderings for proving innermost termination are characterized by the *innermost parallel monotonicity*, *IP-monotonicity* for short. This property may lead to several innermost-specific orderings. Here, an IP-monotonic version of the *Recursive Path Ordering* is presented. This variant can be used (directly or as ingredient of the *Dependency Pairs* method) for proving innermost termination of non-terminating term rewrite systems.

1 Introduction

Rewrite systems are sets of rules used to compute by replacing an instance of the left-hand side of a rule (*redex*) by the corresponding instance of the right-hand side. The replacements are repeated until a term with no redex (*normal form*) is eventually reached. Every replacement (*rewriting step*) involves a non-deterministic choice of both, the redex and the rewriting rule to be applied. Hence, in general one can produce an infinite number of rewriting step sequences started on the same term. A term rewrite system (TRS) is terminating if it has no infinite rewriting sequence.

A common way of restricting the number of rewriting sequences to be inspected when searching for a normal form is to use a rewriting strategy. A TRS can be terminating under a specific strategy whereas not in general. The termination proof for a strategy may be easier and weaker conditions for modularity can be applied. Moreover, for some classes of TRS, proving termination under a particular rewriting strategy suffices for ensuring general termination. Therefore, it turns out to be very important to develop techniques for proving termination of rewriting under strategies.

One of the most commonly used rewriting strategies is the *innermost* one, in which only innermost redexes are reduced. This strategy corresponds to the *"call by value"* computation rule of programming languages and enjoys all the aforementioned advantages. Therefore, studies on properties of innermost rewriting are useful for program verification.

The first and most successful technique for proving innermost termination of rewriting was the *Dependency Pairs* method (DP) [1]. In [20], the *size-change*

* Supported by the project CORDIAL-2, Ref. AML/B7-311-97/0666/II-0021F1A

principle for functional programming [14] was adapted in order to prove innermost termination of rewriting. Moreover, it was combined with DP, obtaining the best of both methods. Other approaches are described in [19, 8, 4]. All these methods are used with general purpose orderings as ingredient, like the *Recursive Path Ordering* (RPO) [6, 12], the Knuth-Bendix Ordering and polynomial interpretations over the reals [3, 15].

In this paper, we study the relationship between innermost termination and well-founded orderings. Stability and monotonicity (which are always required for termination proofs) can be relaxed for termination of this strategy. In innermost rewriting only normalized substitutions are considered. Moreover, very recently it was shown that for innermost termination some monotonicity requirements can be discarded for some function symbols [7]. Here we provide a different approach for relaxing the monotonicity. Our approach was obtained by noting that innermost normalization and termination of the innermost parallel rewriting strategy are equivalent [18]. The latter strategy reduces all innermost redexes of a term at the same time. Therefore, for innermost termination we need to demand monotonicity only after each maximal parallel innermost rewriting step. We call this property *IP-monotonicity* and we show that the suitable orderings for direct innermost termination proofs are IP-monotonic. Another characterization for innermost termination is obtained by combining the innermost parallel relation and DP. As consequence, an innermost termination criterion relying on IP-monotonic quasi-orderings instead of IP-monotonic orderings is also obtained.

IP-monotonicity may lead to new, practical and innermost-specific orderings. In particular, we present an IP-monotonic version of the RPO, called the innermost RPO (iRPO). Its practical application is shown by means of examples. We also show that, for non-overlaying TRSs, the non-strict version of iRPO is an IP-monotonic quasi-ordering. Thus, it can be used as ingredient of DP and effectively combined with the argument filtering method [1, 13].

The rest of the paper is organized as follows. In Section 2 we introduce basic notions and notations. In Section 3 we characterize innermost termination in terms of IP-monotonic (quasi-) orderings. Section 4 is devoted to iRPO and the stability issue.

2 Preliminaries

We assume familiarity with the basics of term rewriting termination (see e.g. [2]).

The set of terms over a signature $\mathcal{F}$ is denoted as $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $\mathcal{X}$ represents a set of variables. Variables are denoted with the letters x, y, z while s, t, u (possibly with subscripts and apostrophes) denote terms. The arity of a function symbol f is denoted as $ar(f)$. The symbol labelling the root of a term t is denoted as $root(t)$. The notation $\bar{t}$ will be ambiguously used to denote either the tuple $(t_1, \ldots, t_n)$ or the multiset $\{t_1, \ldots, t_n\}$.

We assume positions within terms represented by sequences of positive integers, ordered by the prefix ordering. Positions are denoted with the letters p, q (possibly with apostrophes) while for integers we use i, j, k. The *root* position is denoted by λ and $p.q$ denotes the *concatenation* of p and q. The set of positions of a term t is $\mathcal{P}os(t)$. The *subterm* of t at position p is denoted as $t|_p$. The *subterm relation* denoted as $t \rhd t|_p$ in case of $p > \lambda$. The term t with the subterm at position p replaced by s is denoted as $t[s]_p$. Occasionally, we use $t[s]$ to indicate that s is subterm of t.

We say that binary relation $\succ$ is compatible with another binary relation $\succsim$ if $e_1 \succsim e_1' \succ e_2' \succsim e_2$ implies $e_1 \succ e_2$. We call $(\succsim, \succ)$ a *compatible pair* if $\succ$ is well-founded and $\succsim \circ \succ \subseteq \succ$ or $\succ \circ \succsim \subseteq \succ$ [13]. The syntactic equality is denoted as $\equiv$.

Let $\succ$ be an ordering on terms and let $\approx$ be an equivalence relation compatible with $\succ$. The lexicographic extension $\succ^{lex}$ of $\succ$ wrt. $\approx$ for n-tuples is defined as $(s_1, \ldots, s_n) \succ^{lex} (t_1, \ldots, t_n)$ iff $s_1 \approx t_1, \ldots, s_{k-1} \approx t_{k-1}$ and $s_k \succ t_k$ for some $k \in \{1 \ldots n\}$. The extension of $\approx$ to multisets, denoted as $\approx^{mul}$, is the smallest relation s.t. $\emptyset \approx^{mul} \emptyset$ and $S \cup \{s\} \approx^{mul} S' \cup \{t\}$ if $s \approx t \wedge S \approx^{mul} S'$. The extension of $\succ$ to multisets w.r.t. $\approx$ is defined as the smallest ordering $\succ^{mul}$ s.t. $M \cup \{s\} \succ^{mul} N \cup \{t_1, \ldots, t_n\}$ if $M \approx^{mul} N$ and $s \succ t_i$ for all $i \in \{1 \ldots n\}$.

A TRS over $\mathcal{F}$ is denoted as $\mathcal{R}$. The *defined* symbols of $\mathcal{R}$ are $\mathcal{D} = \{root(l) \mid l \to r \in \mathcal{R}\}$. A *rewriting step* with $\mathcal{R}$ is written as $s \to_{\mathcal{R}} t$. The notation $\to_{\mathcal{R},>\lambda}$ is used for a rewriting step at position $p \neq \lambda$. We omit the subscript $\mathcal{R}$ whenever is clear from the context.

A TRS $\mathcal{R}$ is *terminating* if $\to$ is well-founded, i.e. there is no infinite sequence $s_1 \to s_2 \to \ldots$ (sometimes denoted as $s_1 \to^{\infty}$). Alternatively, $\mathcal{R}$ is terminating iff all its rules are included in a reduction ordering [16]. One of the most popular reduction orderings for proving termination is the *Recursive Path Ordering* [6] which is defined below. RPO uses a precedence and can be adapted for dealing with statuses as proposed in [12].

Definition 1. *A* precedence $\succ_{\mathcal{F}}$ *is an ordering on* $\mathcal{F}$ *compatible with an equivalence relation* $\approx_{\mathcal{F}}$*. Let* $\{\mathcal{L}ex, \mathcal{M}ul\}$ *be a partition of* $\mathcal{F}$ *called* statuses *of* $\mathcal{F}$*. The precedence* $\succ_{\mathcal{F}}$ *is compatible with the statuses of* $\mathcal{F}$ *if* $f \approx_{\mathcal{F}} g$ *implies that both* f *and* g *belong to the same part, either* $\mathcal{L}ex$ *or* $\mathcal{M}ul$*.*

Definition 2. *Let* $\succ_{\mathcal{F}}$ *be a precedence over* $\mathcal{F}$ *compatible with the statuses* $\{\mathcal{L}ex, \mathcal{M}ul\}$*. Then* $s = f(\bar{s}) \succ_{rpo} t$ *if one of the following conditions holds:*

1. $s' \succ_{rpo} t$ *or* $s' \approx_{rpo} t$*, for some* $s' \in \bar{s}$
2. $t = g(\bar{t})$*,* $f \succ_{\mathcal{F}} g$ *and* $s \succ_{rpo} t'$ *for all* $t' \in \bar{t}$
3. $t = g(\bar{t})$*,* $f \approx_{\mathcal{F}} g$*,* $f \in \mathcal{F}_{\mathcal{L}ex}$*,* $\bar{s}\ (\succ_{rpo})^{lex}\ \bar{t}$ *and* $s \succ t'$*, for all* $t' \in \bar{t}$*,*
4. $t = g(\bar{t})$*,* $f \approx_{\mathcal{F}} g$*,* $f \in \mathcal{F}_{\mathcal{M}ul}$ *and* $\bar{s}\ (\succ_{rpo})^{mul}\ \bar{t}$*,*

where $s \approx_{rpo} t$ *iff* $s \equiv t$ *or one of the following conditions holds:*

(a) $root(s) \approx_{\mathcal{F}} root(t)$*,* $root(s) \in \mathcal{F}_{\mathcal{L}ex}$ *and* $s_1 \approx_{rpo} t_1, \ldots, s_n \approx_{rpo} t_n$*,*
(b) $root(s) \approx_{\mathcal{F}} root(t)$*,* $root(s) \in \mathcal{F}_{\mathcal{M}ul}$ *and* $\bar{s}(\approx_{rpo})^{mul}\bar{t}$*,*

Theorem 1. *[12]* $\succ_{rpo}$ *is a reduction ordering compatible with the congruence relation* $\approx_{rpo}$*.*

Given a TRS $\mathcal{R}$, $f(t_, \ldots, t_n)$ is said to be *argument normalized* w.r.t. $\mathcal{R}$ if for all $k = 1 \ldots n$, t_k is in normal form w.r.t. $\mathcal{R}$. A pair (s, t) is said to be *argument normalized* if s is so. A *normalized substitution* σ is s.t. $x\sigma$ is in normal form w.r.t. $\mathcal{R}$ for all $x \in \mathcal{D}om(\sigma)$. An *innermost redex* is an argument normalized redex. A term s *rewrites innermost* to t w.r.t. $\mathcal{R}$, written $s \rightarrow_i t$, iff $s \rightarrow t$ at position p and $s|_p$ is an innermost redex. It is said that $\mathcal{R}$ is *innermost terminating* if $\rightarrow_i$ is well-founded.

Example 1. The system $\mathcal{R} = \{g(x, y) \rightarrow x, g(x, y) \rightarrow y, f(0, 1, x) \rightarrow f(x, x, x)\}$ was given by Toyama for proving that termination is not modular for disjoint unions of TRS [21]. This illustrative example has the infinite rewriting sequence: $f(0, 1, g(0, 1)) \rightarrow f(g(0, 1), g(0, 1), g(0, 1)) \xrightarrow{+} f(0, 1, g(0, 1)) \ldots$ However, every innermost rewriting sequence is terminating.

A TRS $\mathcal{R}$ is *innermost confluent* if $\rightarrow_i$ is confluent. We say that $\mathcal{R}$ is *non-overlaying* if there are no two different rules (after renaming variables so that both rules have distinct variables) having unifiable left-hand sides. If $\mathcal{R}$ is non-overlaying, then it is innermost confluent.

3 Characterizing innermost termination of rewriting

In this section we focus on innermost termination, trying to characterize it by means of orderings. The basic idea to achieve this is the fact that all innermost redexes of a term t are in pairwise disjoint positions and moreover, all must be rewritten before reaching a normal form. Hence, if t can be normalized using the innermost strategy, all its innermost redexes can be reduced simultaneously by the parallel innermost strategy [17].

Definition 3. *A term s is reduced innermost in parallel to t w.r.t. $\mathcal{R}$, written $s \mathrel{+\!\!\!\!\!\!\longrightarrow}_i t$, iff $s \xrightarrow{+}_i t$ and either $s \rightarrow_i t$ at position λ or $s = f(\bar{s})$, $t = f(\bar{t})$ and for all $k = 1 \ldots |\bar{s}|$ either $s_k \mathrel{+\!\!\!\!\!\!\longrightarrow}_i t_k$ or $s_k = t_k$ is a normal form.*

It is easy to see that when $s \mathrel{+\!\!\!\!\!\!\longrightarrow}_i t$, t can be obtained by consecutive one-step reductions of all innermost redexes in s. For instance, using the TRS of Example 1 we have $f(g(0, 1), g(0, 1), g(0, 1)) \mathrel{+\!\!\!\!\!\!\longrightarrow}_i f(0, 1, 0)$. The innermost parallel rewrite relation is not only included in the transitive-closure of the innermost rewrite relation but it also characterizes innermost termination. The latter follows from Krishna Rao's contribution concerning the selection invariance for innermost normalization [18]. That is, the choice of innermost redex to be reduced at any step is irrelevant for innermost termination. Thereby, if a TRS is innermost normalizing under a particular strategy then it is innermost normalizing under any other strategy. In order to prove this fact, an oracle based reasoning was used. The following theorem provides a simpler proof for the same result.

Theorem 2. *A TRS $\mathcal{R}$ is innermost terminating iff $\mathrel{+\!\!\!\!\!\!\longrightarrow}_i$ is terminating.*

Proof. The *left-to-right* implication is trivial. For the other direction, it is enough to prove that, for any infinite rewriting sequence $s \to_i^\infty$ there exists an alternate derivation $s \mathrel{+\!\!\!|\!\!\!\to}_i s' \to_i^\infty$.

First, we show that given a derivation $s \xrightarrow{+}_i t$ where t is argument normalized there exists an alternate derivation $s \mathrel{+\!\!\!|\!\!\!\to}_i s' \xrightarrow{*}_i t$, and we do it by structural induction. If the first rewrite step in $s \xrightarrow{+}_i t$ is at position λ, then this derivation is already of the form $s \mathrel{+\!\!\!|\!\!\!\to}_i s' \xrightarrow{*}_i t$. Otherwise, either there is no rewrite step at λ or the first step at λ is on an argument normalized term obtained from s by at least one rewriting step. In any case, the original derivation is of the form $s = f(\bar{s}) \xrightarrow{+}_{i,>\lambda} f(\bar{t}) \xrightarrow{*}_i t$, where $f(\bar{t})$ is argument normalized and every $s_k \in \bar{s}$ is either a normal form and we call $s'_k = s_k$, or $s_k \xrightarrow{+}_i t_k$ and by induction hypothesis $s_k \mathrel{+\!\!\!|\!\!\!\to}_i s'_k \xrightarrow{*}_i t_k$. Therefore, $s = f(\bar{s}) \mathrel{+\!\!\!|\!\!\!\to}_i f(\bar{s'}) \xrightarrow{*}_i t$, as desired.

Now, given a derivation $s \to_i^\infty$, we show that there exists an alternate derivation $s \mathrel{+\!\!\!|\!\!\!\to}_i s' \to_i^\infty$ by structural induction. If the first rewrite step is at λ position, the result trivially holds. Otherwise, if there is some rewrite step at λ, then this derivation is of the form $s \xrightarrow{+}_i t \to_i^\infty$ where t is argument normalized, and by our previous statement, there exists an alternate derivation $s \mathrel{+\!\!\!|\!\!\!\to}_i s' \to_i^* t \to_i^\infty$, and the result holds. If there is no rewrite step at λ in $s \to_i^\infty$, then s is of the form $f(\bar{s})$ and for some $s_k \in \bar{s}$, say s_1, there exists an infinite rewriting sequence $s_1 \to_i^\infty$. By induction hypothesis, there exists an alternate derivation $s_1 \mathrel{+\!\!\!|\!\!\!\to}_i s'_1 \to_i^\infty$. For the rest of s_k's, either s_k is a normal form and we call $s'_k = s_k$, or a parallel innermost rewriting step can be applied on s_k, i.e. $s_k \mathrel{+\!\!\!|\!\!\!\to}_i s'_k$ for some s'_k. Therefore, there exists an alternate derivation $s = f(\bar{s}) \mathrel{+\!\!\!|\!\!\!\to}_i f(\bar{s'}) \to_i^\infty$, and the result follows. □

This theorem leads us to define the *innermost parallel monotonicity*, *IP-monotonicity* for short, directly from $\mathrel{+\!\!\!|\!\!\!\to}_i$.

Definition 4. *A binary relation $\succ$ is IP-monotonic w.r.t. a TRS $\mathcal{R}$ iff $\mathrel{+\!\!\!|\!\!\!\to}_i \subseteq \succ$.*

The IP-monotonicity hides a weak kind of stability and monotonicity. This can be seen in the next lemma, which is a straightforward conclusion from Definition 4.

Lemma 1. *A binary relation $\succ$ is IP-monotonic w.r.t. $\mathcal{R}$ iff*

- *$l\sigma \succ r\sigma$ for all $l \to r \in \mathcal{R}$ and substitution σ s.t. $l\sigma$ is argument normalized and*
- *$\bar{s} \mathrel{+\!\!\!|\!\!\!\to}_i \bar{t}$ implies $f(\bar{s}) \succ f(\bar{t})$ for all $f \in \mathcal{F}$.*

Using this lemma is easy to see that any transitive, monotonic and stable binary relation including $\mathcal{R}$ is also in-monotonic w.r.t. $\mathcal{R}$. Therefore, reduction orderings suffices for innermost termination. However, termination of this strategy is indeed characterized by IP-monotonic and well-founded orderings.

Theorem 3. *A TRS $\mathcal{R}$ is innermost terminating iff there is a well-founded relation $\succ$ which is IP-monotonic w.r.t. $\mathcal{R}$.*

Proof. The left-to-right implication can be easily shown by taking $\overset{+}{\Vert\!\!\!\rightarrow}_i$. For the converse, if $\mathcal{R}$ is not innermost terminating, by Theorem 2, there exists an infinite rewriting sequence $s_1 \Vert\!\!\!\rightarrow_i s_2 \Vert\!\!\!\rightarrow_i \ldots$. By IP-monotonicity of $\succ$ w.r.t. $\mathcal{R}$, $s_1 \succ s_2 \succ \ldots$, contradicting the well-foundedness of $\succ$. $\square$

In the context of DP, innermost termination was characterized through the use of chains. Given a TRS $\mathcal{R}$, $\langle f(\bar{s}), g(\bar{t})\rangle$ is a *dependency pair* of $\mathcal{R}$ if $f(\bar{s}) \rightarrow u[g(\bar{t})] \in \mathcal{R}$ and $g \in \mathcal{D}$[1]. The set of all dependency pairs of $\mathcal{R}$ is denoted as $\mathcal{DP}(\mathcal{R})$. A sequence of dependency pairs $S = \langle s_1, t_1\rangle\ \langle s_2, t_2\rangle\ \langle s_3, t_3\rangle \ldots$ of $\mathcal{R}$ is an innermost $\mathcal{R}$-chain if there is a substitution σ s.t. for all $j > 0$, $s_j\sigma$ is argument normalized and $t_j\sigma \overset{*}{\rightarrow}_i s_{j+1}\sigma$ holds. A TRS $\mathcal{R}$ is innermost terminating iff there is no infinite innermost $\mathcal{R}$-chain [1].

Since in every innermost $\mathcal{R}$-chain $s_j\sigma$ is argument normalized , by the proof of Theorem 2, we have $t_j\sigma \overset{*}{\Vert\!\!\!\rightarrow}_i s_{j+1}\sigma$, for all $j > 0$. Therefore, the parallel innermost relation can also be used for characterizing innermost termination by means of chains. Furthermore, we can use a compatible pair $(\succsim, \succ)$ s.t. $\succsim$ is IP-monotonic w.r.t. $\mathcal{R}$.

Theorem 4. *A TRS $\mathcal{R}$ is innermost terminating iff there is a compatible pair $(\succsim, \succ)$ s.t. $\succsim$ is IP-monotonic w.r.t. $\mathcal{R}$ and $s\sigma \succ rt\sigma$ for all $\langle s, t\rangle \in \mathcal{DP}(\mathcal{R})$ and substitution σ s.t. $s\sigma$ is argument normalized.*

Proof. For the *right-to-left* direction suppose $\mathcal{R}$ is not innermost terminating. Then, there is an infinite innermost $\mathcal{R}$-chain $\langle s_1, t_1\rangle\ \langle s_2, t_2\rangle\ \langle s_3, t_3\rangle \ldots$ and a substitution σ s.t. for all $j > 0$, $s_j\sigma$ is argument normalized and $t_j\sigma \overset{*}{\rightarrow}_i s_{j+1}\sigma$. Since $t_j\sigma \overset{*}{\Vert\!\!\!\rightarrow}_i s_{j+1}\sigma$ and $\succsim$ is IP-monotonic, we have $t_j\sigma\ (\succsim \cup \equiv)\ s_{j+1}\sigma$. Besides $s_j\sigma \succ t_j\sigma$ holds by assumption. Hence, seeing that $\succsim \circ \succ \subseteq \succ$ or $\succ \circ \succsim \subseteq \succ$, we obtain the infinite sequence $s_1\sigma \succ s_2\sigma \succ s_3\sigma \succ \ldots$ which contradicts the well-foundedness of $\succ$.

For the *left-to-right* direction we take $\succsim = \succ = (\rightarrow_i \cup \rhd)^+$. Clearly, $\succ \circ \succ \subseteq \succ$, $\succ$ is IP-monotonic w.r.t. $\mathcal{R}$ and orients $\mathcal{DP}(\mathcal{R})$. Finally, when $\mathcal{R}$ is innermost terminating, $\overset{+}{\rightarrow}_i$ is a monotonic and well-founded ordering and thereby $(\rightarrow_i \cup \rhd)^+$ is also well-founded. $\square$

4 An example of IP-monotonic ordering

Multiset extensions have been used for defining successful reduction orderings like RPO and MSPO [5]. This is because they preserve suitable properties like irreflexivity, transitivity, stability and well-foundedness. Besides, every ordering

[1] The original notion of dependency pair is $\langle \widehat{f}(\bar{s}), \widehat{g}(\bar{t})\rangle$ where $\widehat{f}$ and $\widehat{g}$ are *marked* (or *tuple*) symbols associated to f and g resp. This renaming allows to apply a different treatment to function symbols when they appear on top of dependency pairs. We have chosen the unmarked version for simplicity but using the marked version does not affect our results.

$\succ$ is monotonic on $\succ^{mul}$ in the sense that $s \succ t$ implies $\{s_1, \ldots, s, \ldots, s_n\} \succ^{mul} \{s_1, \ldots, t, \ldots, s_n\}$. Once the terms of a multiset are rewritten with $\dashrightarrow\!\!\!\!\!\!+_i$ w.r.t. a TRS $\mathcal{R}$, all reducible terms decrease w.r.t. every IP-monotonic ordering $\succ$ whereas normal forms remain untouched. Therefore, the original multiset also decreases w.r.t. $\succ^{mul}$. Even more, the comparison with $\succ^{mul}$ still holds if we remove all multiple occurrences from the original and the reduced multisets.

Based on this fact we adapt (actually extend) RPO for proving innermost termination. This is achieved just by adding a new status $\mathcal{F}_{\mathcal{S}et}$ which allows certain terms to be compared using the *set* (instead of the multiset) of their arguments. The new ordering is the first which is innermost-specific; therefore we call it the *innermost Recursive Path Ordering*. Its definition can be formulated either by cases like RPO or by transformation, i.e. first we eliminate repetitions and then compare with RPO. The latter alternative provides an elegant definition and straightforward proofs for iRPO's properties.

Definition 5. *Given $\mathcal{F}_{\mathcal{S}et} \subseteq \mathcal{F}$, the transformation ϕ over $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is defined as*

- $\phi(x) = x$*, if* $x \in \mathcal{X}$
- $\phi(f(s_1, \ldots, s_n)) = f(\phi(s_1), \ldots, \phi(s_n))$*, if* $f \notin \mathcal{F}_{\mathcal{S}et}$
- *otherwise* $\phi(f(s_1, \ldots, s_n)) = f(\phi(s_{j_1}), \ldots, \phi(s_{j_m}))$ *where* $j_1 < \ldots < j_m$ *are the* j*'s in* $\{1 \ldots n\}$ *s.t.* $s_k \not\equiv s_j$ *for all* $k < j$*. In other words, the tuple* $(s_{j_1} \ldots s_{j_m})$ *is just* $(s_1 \ldots s_n)$ *after removing repetitions from left to right.*

Given an RPO ordering $\succ_{rpo}$ and $\mathcal{F}_{\mathcal{S}et} \subseteq \mathcal{F}_{\mathcal{M}ul}$, the corresponding $\succ_{irpo}$ ordering is defined as $s \succ_{irpo} t$ iff $\phi(s) \succ_{rpo} \phi(t)$. If $\approx_{rpo}$ is the equivalence relation corresponding to $\succ_{rpo}$, then $\approx_{irpo}$ is defined as $s \approx_{irpo} t$ iff $\phi(s) \approx_{rpo} \phi(t)$. The union of $\succ_{irpo}$ and $\approx_{irpo}$ is denoted as $\succsim_{irpo}$.

Note that after applying ϕ, some symbols in $\mathcal{F}_{\mathcal{S}et}$ may become varyadic. Besides, repetitions are removed before applying ϕ, not later. For example, if $\mathcal{F}_{\mathcal{S}et} = \{h, g\}$, $\phi(h(g(a, b, b, a), g(a, b, c, b), g(a, b, b, c))) = h(g(a, b), g(a, b, c), g(a, b, c))$. Although ϕ removes repetitions from left to right, any other fixed order would give the same definition of $\succ_{irpo}$ and $\approx_{rpo}$ above. Clearly, the transformed terms might be different (for instance, choosing the right-to-left order $\phi(g(a, b, c, b)) = g(a, c, b)$). But this is irrelevant since the multiset comparison is used for comparing the affected arguments.

The following proposition is a direct consequence of the definition of $\approx_{irpo}$, $\succ_{irpo}$ and Theorem 1.

Proposition 1. *$\succ_{irpo}$ is a well-founded ordering compatible with the equivalence relation $\approx_{irpo}$.*

Now, we show that if the set of argument normalized instances of a TRS $\mathcal{R}$ can be oriented using iRPO then the ordering is IP-monotonic w.r.t. $\mathcal{R}$. Therefore, by Theorem 3 and Proposition 1, it can be used for proving innermost termination.

Theorem 5. *$\succ_{irpo}$ is IP-monotonic w.r.t. a TRS $\mathcal{R}$ iff $l\sigma \succ_{irpo} r\sigma$, for every rule $l \to r \in \mathcal{R}$ and substitution σ s.t. $l\sigma$ is argument normalized.*

Proof. The *left-to-right* implication follows by definition of IP-monotonicity. For the other direction we need to show that $s \mathrel{+\!\!\!\!+\!\!\!\!\to}_i t$ implies $s \succ_{irpo} t$. By assumption, $s \succ_{irpo} t$ whenever $s \mathrel{+\!\!\!\!+\!\!\!\!\to}_i t$ at position λ. Otherwise $s = f(\bar{s})$, $t = f(\bar{t})$ and for all $k \in \{1 \ldots |\bar{s}|\}$ either $s_k = t_k$ is a normal form, or $s_k \mathrel{+\!\!\!\!+\!\!\!\!\to}_i t_k$ and using structural induction we have $s_j \succ_{irpo} t_j$, and hence $\phi(s_j) \succ_{rpo} \phi(t_j)$. Moreover, for some $k \in \{1 \ldots |\bar{s}|\}$, s_k is not a normal form, and consequently $s_k \succ_{irpo} t_k$ and $\phi(s_k) \succ_{rpo} \phi(t_k)$. Now, if $f \notin \mathcal{F}_{\mathcal{S}et}$, then $s \succ_{irpo} t$ by monotonicity and transitivity of $\succ_{rpo}$. Otherwise, let $\phi(s) = f(s'_1, \ldots, s'_m)$, $\phi(t) = f(t'_1, \ldots, t'_n)$. Moreover, let S, T and S' be the multisets $\{s'_1, \ldots, s'_m\}$, $\{t'_1, \ldots, t'_n\}$ and $S' = \{\phi(s_k) \mid k \in \{1 \ldots |\bar{s}|\}, s_k$ is a normal form w.r.t $\mathcal{R}\}$[2] respectively. Then $S' \subset S$, $S' \subseteq T$ and for all $v \in T - S'$ there is some $u \in S - S'$ s.t. $u \succ_{rpo} v$ holds by induction. Therefore, $\{s'_1, \ldots, s'_m\} \succ_{rpo}^{mul} \{t'_1, \ldots, t'_n\}$ by definition of the multiset extension, and $\phi(s) \succ_{rpo} \phi(t)$ holds. □

Example 2. Toyama's TRS (see Example 1) can be shown innermost terminating using iRPO. For the first two rules, $g(x,y)\sigma \succ_{rpo} x\sigma$ and $g(x,y)\sigma \succ_{rpo} y\sigma$ hold by case 1 for every substitution σ. Moreover, every instance of the last rule can be oriented by defining $\mathcal{F}_{\mathcal{S}et} = \mathcal{F}_{\mathcal{M}ul} = \{f\}$. Note that depending on the value of $x\sigma$ we have the following situations, all of them holding by case 3.

1. if $x\sigma = 0$ then $\phi(f(0,1,x)\sigma) = f(0,1) \succ_{rpo} f(0) = \phi(f(x,x,x)\sigma)$,
2. if $x\sigma = 1$ then $\phi(f(0,1,x)\sigma) = f(0,1) \succ_{rpo} f(1) = \phi(f(x,x,x)\sigma)$,
3. otherwise $\phi(f(0,1,x)\sigma) = f(0,1,x\sigma) \succ_{rpo} f(x\sigma) = \phi(f(x,x,x)\sigma)$.

Since the transformation ϕ unites duplicated arguments, other Toyama-like examples can be included in iRPO (e.g. [1, Examples 5.2.3,5.2.13,5.2.14]). When such multiple occurrences appear at top level, the techniques for cancelling cycles in the estimated innermost dependency graph [1, 4, 9] also handle many of these systems. However, as the next example shows, the latter does not hold in general.

Example 3. The next TRS is a more complex variant of Toyama's example.

$$\mathcal{R}_1 = \begin{cases} f(x,x,y) \to h(y) \\ h(x) \to x \\ h(f(x,y,z)) \to f(z,z,y) \\ h(f(x,y,z)) \to f(y,y,x) \\ c(f(0,1,x),x) \to c(f(x,x,x),h(x)) \end{cases}$$

This system has the following infinite sequence (the redex used in each rewriting step appears underlined):

[2] We construct S' by selecting just one occurrence of every normal form in $\bar{s}$. Note that S, T and even S' may have repeated elements, because ϕ is not injective.

$$
\begin{array}{l}
c(f(0,1,h(f(0,1,0))),h(f(0,1,0))) \ \rightarrow \\
c(f(h(f(0,1,0)),h(f(0,1,0)),h(f(0,1,0))),h(h(f(0,1,0)))) \ \rightarrow \\
c(f(f(1,1,0),h(f(0,1,0)),h(f(0,1,0))),h(h(f(0,1,0)))) \ \rightarrow \\
c(f(f(1,1,0),f(0,0,1),h(f(0,1,0))),h(h(f(0,1,0)))) \ \rightarrow \\
c(f(f(1,1,0),f(0,0,1),h(f(0,1,0))),h(f(0,1,0))) \ \rightarrow \\
c(f(h(0),f(0,0,1),h(f(0,1,0))),h(f(0,1,0))) \ \rightarrow \\
c(f(0,f(0,0,1),h(f(0,1,0))),h(f(0,1,0))) \ \rightarrow \\
c(f(0,h(1),h(f(0,1,0))),h(f(0,1,0))) \ \rightarrow \\
c(f(0,1,h(f(0,1,0))),h(f(0,1,0))) \ \rightarrow \ldots
\end{array}
$$

However, $\mathcal{R}_1$ is indeed innermost terminating. Proving this fact automatically is hard to obtain with the existing methods. Obviously, no termination technique can be used in this case. Furthermore, the estimations for the innermost dependency graph do not cancel the problematic cycle corresponding to the last rule. The use of polynomials with negative coefficients has been proposed for innermost termination proofs [1, 7]. But the practical results concerning the automated generation of such polynomials are still few, and for instance the method described in [11] cannot be applied to this system.

Nevertheless, innermost termination of $\mathcal{R}_1$ can be proved using iRPO with $\mathcal{F}_{\mathcal{S}et} = \mathcal{F}_{\mathcal{M}ul} = \{f\}$, $\mathcal{F}_{\mathcal{L}ex} = \{c\}$ and the precedence $c \succ_{\mathcal{F}} h$ and $f \succ_{\mathcal{F}} h$. Every instance of the first rule decreases by case 2. For the next three rules, $l\sigma \succ_{irpo} r\sigma$ holds by case 1, for every substitution σ (note that $f(x,y,z)\sigma \succsim_{irpo} f(z,z,y)\sigma$). Finally, considering the situations of the previous example the last rule is easily oriented using case 4.

4.1 Innermost Stability for iRPO

Theorem 5 is not suitable for automation since one has to check infinitely many instantiation of the rules. Hence, though stability is not necessary for innermost termination, it is always a desirable property when proving termination.

Unlike RPO, iRPO is not stable. The problem comes from the fact that two different terms can be equal after applying a substitution. Thereby, in general $\phi(s\sigma) \neq \phi(s)\sigma$ when $s \rhd t$ and $root(t) \in \mathcal{F}_{\mathcal{S}et}$. For example, for $\mathcal{F}_{\mathcal{S}et} = \{f\}$ and $\sigma = \{y \mapsto x\}$ we have $\phi(f(c(x),c(y))\sigma) = f(c(x)) \neq \phi(f(c(x),c(y)))\sigma$ $f(c(x),c(x))$. Due to this $\succ_{irpo}$ is not stable. For example, $f(c(x),c(y)) \succ_{irpo}$ $f(c(x),c(x))$ and $f(c(x),c(y)) \succ_{irpo} f(x,c(x))$ hold but do not after applying the former substitution. Note that $\phi(f(c(x),c(y))\sigma) = f(c(x)) = \phi(f(c(x),c(x))\sigma)$ and $\phi(f(c(x),c(y))\sigma) = f(c(x)) \not\succ_{rpo} f(x,c(x)) = \phi(x,f(c(x))\sigma)$.

Definition 6. *The problem of* iRPO stability *is defined as follows.*

Instance: *two terms s and t, a term rewrite system $\mathcal{R}$, and an iRPO ordering $\succ_{irpo}$.*
Question: *Is $s\sigma \succ_{irpo} t\sigma$ for any substitution σ such that $s\sigma$ is argument normalized?*

As we will see, this problem is co-NP-complete. The following algorithm non-deterministically decides the complement of the iRPO stability problem, i.e., if there exists a substitution σ such that $s\sigma$ is argument normalized and $s\sigma \not\succ_{irpo} t\sigma$ for given terms s and t.

Algorithm 1

1. Let $E_p \subseteq \{(i,j) \mid 1 \leq i < j \leq ar(root(s|_p))\}$ be an selection of pairs for every position $p \in \mathcal{P}os_{\mathcal{F}_{Set}}(s)$.
2. Let σ be the m.g.u. of the set of equations $\{s|_{p.i} = s|_{p.j} \mid p \in \mathcal{P}os_{\mathcal{F}_{Set}}(s), (i,j) \in E_p\}$. Check if $s\sigma$ is argument normalized w.r.t. $\mathcal{R}$ and $s\sigma \not\succ_{irpo} t\sigma$, and give this result as output.

Lemma 2. *The Algorithm* 1 *non-deterministically decides the complement of the iRPO stability problem.*

Before giving the proof of the previous lemma, we will need the following two technical results.

Lemma 3. *Let σ be the m.g.u. of a set of equations S. For every term s occurring in S and for every position $p \in \mathcal{P}os_{\mathcal{F}}(s\sigma)$, there exists a term s' occurring in S and a position $p' \in \mathcal{P}os_{\mathcal{F}}(s')$ such that $s\sigma|_p \equiv s'\sigma|_{p'}$.*

Proof. This can easily be proved by induction on the number of steps of many known unification algorithms. In those algorithms, the m.g.u. σ is incrementally obtained by, first, making σ_0 to be the identity substitution. Then, at some step, it is modified by an assignment of the form $\sigma_{i+1} := \sigma_i\{x \mapsto t\sigma_i\}$, where t is a term occurring in S, $x\sigma_i = x$, and any variable y occurring in $t\sigma_i$ satisfies that x does not occur in $y\sigma_i$. It is not difficult to see that, if σ_i satisfies the condition of the lemma, then σ_{i+1} does. □

Lemma 4. *For every term s and substitution σ we have $\phi(s)\phi(\sigma) \succsim_{rpo} \phi(s\sigma)$* [3]. *Moreover, if for every pair of positions $p.i$ and $p.j$ of s with $p \in \mathcal{P}os_{\mathcal{F}_{Set}}(s)$ it holds that $(s|_{p.i} \equiv s|_{p.j}) \Leftrightarrow (s|_{p.i}\sigma \equiv s|_{p.j}\sigma)$, then $\phi(s)\phi(\sigma) \equiv \phi(s\sigma)$*

Proof. Clearly $\phi(s)\phi(\sigma) \succsim_{rpo} \phi(s\sigma)$ holds since $\phi(s\sigma)$ can be obtained from $\phi(s)\phi(\sigma)$ by eventually removing some subterms at positions below a symbol with multiset status. Now, assume that for every pair of positions $p.i$ and $p.j$ of $\mathcal{P}os(s)$, it holds that $(s|_{p.i} \equiv s|_{p.j}) \Leftrightarrow (s|_{p.i}\sigma \equiv s|_{p.j}\sigma)$. Proving $\phi(s)\phi(\sigma) \equiv \phi(s\sigma)$, is equivalent to see that any position $p.i$ with $p \in \mathcal{P}os_{\mathcal{F}_{Set}}(s)$ satisfies that for all j in $1 \ldots i-1$, $s|_{p.j} \equiv s|_{p.i}$ if and only if $s|_{p.j}\sigma \equiv s|_{p.i}\sigma$ (i.e. the removing action of ϕ coincides on s and $s\sigma$ at positions in $\mathcal{P}os(s)$). But this is trivial by our assumption. □

Now, we are ready to prove Lemma 2.

[3] Here, ϕ is adapted to substitutions in a natural way, i.e. $x\phi(\sigma) = \phi(x\sigma)$.

Proof. (Of Lemma 2) If the algorithm gives a positive answer, then it is clear that there exists a substitution σ (the one obtained by the algorithm) satisfying that $s\sigma$ is argument normalized and $s\sigma \not\succ_{irpo} t\sigma$.

Hence, it remains to see that, if for some substitution σ, $s\sigma$ is argument normalized and $s\sigma \not\succ_{irpo} t\sigma$, then there is a selection E_p for every $p \in \mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s)$ that produces a positive answer. The selection we need for every of such p's is $E_p = \{(i,j) \mid s\sigma|_{p.i} \equiv s\sigma|_{p.j}\}$. Let σ' be the m.g.u. of the corresponding set of equations S in the algorithm. We have that $\sigma = \sigma'\sigma''$ for some σ'' (since σ is an unifier of S), and that for all $p \in \mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s)$, $s\sigma'|_{p.i} \equiv s\sigma'|_{p.j}$ if and only if $s\sigma'|_{p.i}\sigma'' \equiv s\sigma'|_{p.j}\sigma''$. Moreover, $s\sigma'$ is argument normalized since $s\sigma'\sigma'' = s\sigma$ is. It remains to see that $s\sigma' \not\succ_{irpo} t\sigma'$, or, equivalently, that $\phi(s\sigma') \not\succ_{rpo} \phi(t\sigma')$. We do it by contradiction, i.e. assume that $\phi(s\sigma') \succ_{rpo} \phi(t\sigma')$. By stability of $\succ_{rpo}$, it holds that $\phi(s\sigma')\phi(\sigma'') \succ_{rpo} \phi(t\sigma')\phi(\sigma'')$. By the first part of Lemma 4, $\phi(t\sigma')\phi(\sigma'') \succsim_{rpo} \phi(t\sigma'\sigma'') \equiv \phi(t\sigma)$. If we could prove $\phi(s\sigma')\phi(\sigma'') \equiv \phi(s\sigma'\sigma'')$ then we would obtain $\phi(s\sigma) \succ_{rpo} \phi(t\sigma)$, and hence, $s\sigma \succ_{irpo} t\sigma$, which is a contradiction with our assumption.

In order to prove $\phi(s\sigma')\phi(\sigma'') \equiv \phi(s\sigma'\sigma'')$, we want to apply the second part of Lemma 4. We already know that for all $p \in \mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s)$, $s\sigma'|_{p.i} \equiv s\sigma'|_{p.j}$ if and only if $s\sigma'|_{p.i}\sigma'' \equiv s\sigma'|_{p.j}\sigma''$. It remains to see that this property extends to $s\sigma'$, i.e., for all $p \in \mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s\sigma')$, $s\sigma'|_{p.i} \equiv s\sigma'|_{p.j}$ if and only if $s\sigma'|_{p.i}\sigma'' \equiv s\sigma'|_{p.j}\sigma''$. For this goal, it is enough to see that for any position $p \in \mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s\sigma')$, there exists a position $p' \in \mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s)$ such that $s\sigma'|_p \equiv s\sigma'|_{p'}$. But this is easy by means of Lemma 3 as follows. First, note that if instead of considering the set of equations S we consider $S \cup \{s = s\}$, then σ' continues being the m.g.u. of this set. Now, let p be a position in $\mathcal{P}os_{\mathcal{F}_{\mathcal{S}et}}(s\sigma')$. By Lemma 3, there exists a term s' in $S \cup \{s = s\}$ and a non-variable position p' in s' such that $s\sigma'|_p \equiv s'\sigma'|_{p'}$. But this term s' can be considered to be s, since all terms occurring in S are subterms of s. □

Theorem 6. *The iRPO stability problem is co-NP-complete*

Proof. Since we have proved the correctness of the Algorithm 1, for seeing that the complement of this problem belongs to NP, it only remains to see that such an algorithm takes polynomial time. The selection E_p for every p and the corresponding set of equations need polynomial time. A most general unifier σ can be represented in polynomial space on the given set of equations by means of DAG's, and computed in polynomial time. Checking the irreducibility of $s\sigma$, obtaining the DAG's representing $\phi(s\sigma)$ and $\phi(t\sigma)$, and checking if $\phi(s\sigma) \not\succ_{rpo} \phi(t\sigma)$ takes polynomial time as well.

For proving that the complement is an NP-hard problem we give a reduction from 3-SAT. Given an instance of 3-SAT with variables $x_1 \dots x_n$ and clauses $c_1 \dots c_m$, we construct the following terms s and t based on the signature $\{h, f, g, g', 0, 1\}$ where h and f have lexicographic status and arity 2, g and g' have set status and arities 4 and 5 respectively, and 0 and 1 are constants. In s and t appear the (term) variables $x_1, \dots, x_n, \overline{x_1}, \dots, \overline{x_n}$.

$$s = h(\ f(v_1, f(v_2 \dots, f(v_{n-1}, v_n) \dots))\ ,\ f(u_1, f(u_2 \dots, f(u_{m-1}, u_m) \dots))\ )$$

$$t = h(\ f(v'_1, f(v'_2 \ldots, f(v'_{n-1}, v'_n)\ldots))\ ,\ f(u'_1, f(u'_2 \ldots, f(u'_{m-1}, u'_m)\ldots))\ )$$

where $v_i = g(x_i, \overline{x_i}, 0, 1)$, $v'_i = g(x_i, \overline{x_i}, x_i, \overline{x_i})$ and if c_i is a clause with literals l_j, l_k, l_o, then $u_i = g'(l_j, l_k, l_o, 0, 1)$ and $u'_i = g'(l_j, l_k, l_o, 0, 0)$.

Regardless the precedence, it is easy to see that there exists σ satisfying $s\sigma \not\succ_{irpo} t\sigma$ if and only if the original 3-SAT problem is satisfiable. Note that, since $g \in \mathcal{F}_{\mathcal{S}et}$, the term $g(x_i, \overline{x_i}, 0, 1)\sigma$ is not greater than $g(x_i, \overline{x_i}, x_i, \overline{x_i})\sigma$ only if σ assigns 0 and 1, or 1 and 0, to the variables x_i and $\overline{x_i}$, respectively. Besides, the term $g'(l_j, l_k, l_o, 0, 1)\sigma$ is not not greater than $g'(l_j, l_k, l_o, 0, 0)\sigma$ only if σ satisfies every clause c_i with literals l_j, l_k and l_o. By considering an empty $\mathcal{R}$ the result follows. □

4.2 Using iRPO for DP

In general, the compatible pair $(\succsim_{irpo}, \succ_{irpo})$ cannot be used for proving innermost termination with DP. This is because $\succsim_{irpo}$ is not IP-monotonic w.r.t. an arbitrary TRS $\mathcal{R}$. Unlike iRPO, the latter holds even if $\succsim_{irpo}$ orients every rule instance whose left-hand side is argument normalized. Note that after an innermost parallel step, it may happen that an occurrence of a duplicated argument of a symbol in $\mathcal{F}_{\mathcal{S}et}$ decreases w.r.t. $\succ_{irpo}$ while another occurrence remains equal w.r.t. $\approx_{irpo}$. Hence, the corresponding set of arguments may neither decrease w.r.t $(\succ_{irpo})^{mul}$ nor remain equal w.r.t. $(\approx_{irpo})^{mul}$.

As an alternative, we may combine the argument filtering technique [1, 13] with $\succ_{irpo}$ in order to obtain a compatible pair. An argument filtering over a signature $\mathcal{F}$ is a function π s.t. for all $f \in \mathcal{F}$, either $\pi(f) \in \{1 \ldots ar(f)\}$ or $\pi(f) \subseteq \{1 \ldots ar(f)\}$. It induces a mapping from $\mathcal{T}(\mathcal{F}, \mathcal{X})$ to $\mathcal{T}(\mathcal{F}_\pi, \mathcal{X})$ as follows:

$$\begin{cases} \pi(x) = x & if \;\; x \in \mathcal{X} \\ \pi(f(t_1, \ldots, t_n)) = \pi(t_i) & if \;\; \pi(f) = i, \\ \pi(f(t_1, \ldots, t_n)) = f(\pi(t_{i_1}), \ldots, \pi(t_{i_m})) & if \;\; \pi(f) = [i_1, \ldots, i_m], \end{cases}$$

where $[i_1, \ldots, i_m]$ denotes an ordered set and $\mathcal{F}_\pi$ consists of all symbols f s.t. $\pi(f)$ is a set (the arity of every $f \in \mathcal{F}_\pi$ is $|\pi(f)|$). Given a set of pairs $\mathcal{P}$, $\pi(\mathcal{P})$ denotes $\{(\pi(s), \pi(t)) \mid (s, t) \in \mathcal{P}\}$.

Given a binary relation $\succ$, the relation $\succ_\pi$ is defined as $s \succ_\pi t$ iff $\pi(s) \succ \pi(t)$. It not difficult to see that when $\succ$ is monotonic (resp. stable) we have that $s \succ_\pi t$ implies $u[s] \succeq_\pi u[t]$ (resp. $s\sigma \succ_\pi t\sigma$). Therefore, this method has been used for obtaining a monotonic quasi-ordering from a monotonic ordering while preserving stability and well-foundedness.

The iRPO ordering is already defined via a transformation: $s \succ_{irpo} t$ iff $\phi(s) \succ_{rpo} \phi(t)$. Hence, two possibilities seem natural to be considered for combining it with an argument filtering π: we can compare two terms s and t by either $\phi(\pi(s)) \succ_{rpo} \phi(\pi(t))$ or $\pi(\phi(s)) \succ_{rpo} \pi(\phi(t))$. Applying π before ϕ does not work well. An argument filtering might transform a redex into a filtered normal form. Hence, since the transformation ϕ removes duplicated arguments, some innermost parallel reductions might be lost, i.e. it might happen that $s \mathrel{+\!\!\!\Vdash\!\!\to_i} t$ but $\pi(s) \not\succeq \pi(t)$. The next example illustrates this situation.

Example 4. The following non-overlapping system is not innermost terminating.

$$\mathcal{R}_2 = \begin{cases} h(0) \to 0 \\ h(1) \to 1 \\ f(0,1,h(2)) \to f(h(0),h(1),h(2)) \end{cases}$$

If we remove the argument of h then we obtain the ordering constraints $h \succ 0, h \succ 1, f(0,1,h) \succ f(h,h,h)$. These constraints are satisfied by $\succ_{irpo}$ with $\mathcal{F}_{\mathcal{S}et} = \{f\}$, $h \succ_{\mathcal{F}} 0$ and $h \succ_{\mathcal{F}} 1$. Therefore, one could falsely prove (innermost) termination of $\mathcal{R}_2$.

Note that $f(h(0),h(1),h(2)) = s \mathrel{+\!\!\!\!\!\mapsto}_i t = f(0,1,h(2))$ and even $s \succ_{irpo} t$ but since $\{h\} \not\succ^{mul}_{irpo} \{0,1,h\}$, we have $f(h,h,h) = \pi(s) \not\succ_{irpo} \pi(t) = f(0,1,h)$.

Hence, we consider the other possibility, i.e. to apply the transformation ϕ before the filtering π. In this case, it is natural to demand that π does not affect the symbols in $\mathcal{F}_{\mathcal{S}et}$ (i.e. $\pi(f(t_1,\ldots,t_n)) = f(\pi(t_1),\ldots,\pi(t_n))$, for all $f \in \mathcal{F}_{\mathcal{S}et}$), since some arguments might be previously removed by ϕ. In general, this approach does not work either.

Example 5. The TRS $\mathcal{R}_3 = \{a \to b, a \to c, g(c) \to d, f(g(b),d) \to f(g(a),g(a))\}$ is not innermost terminating. But taking $\mathcal{F}_{\mathcal{S}et} = \{f\}$, $\pi(g) = \emptyset$ and the precedence $a \succ_{\mathcal{F}} b, a \succ_{\mathcal{F}} c, g \succ_{\mathcal{F}} d$, we have $\pi(\phi(\mathcal{R}_3)) \subset \succ_{rpo}$. Hence, one may erroneously conclude $\mathcal{R}_3$ is innermost terminating.

Nevertheless, for non-overlaying TRSs this approach indeed yields the desired result. Non-overlayingness is not a very restrictive condition for a TRS in the context of innermost rewriting. This strategy corresponds to the usual behavior of programming languages, where arguments are fully evaluated before applying a function. If a program is deterministic, which is the usual situation, then it corresponds to a non-overlaying system. Besides, this family of TRSs includes non-overlapping ones for which termination and innermost termination coincide [10].

Definition 7. *Let $\succ_{rpo}$ be an RPO ordering with $\mathcal{F}_{\mathcal{S}et} \subseteq \mathcal{F}_{\mathcal{M}ul}$ and π be an argument filtering over $\mathcal{F} - \mathcal{F}_{\mathcal{S}et}$. The corresponding $\succ_{irpo,\pi}$ ordering is defined as $s \succ_{irpo,\pi} t$ iff $\pi(\phi(s)) \succ_{rpo} \pi(\phi(t))$. If $\approx_{rpo}$ is the equivalence relation corresponding to $\succ_{rpo}$, then $\approx_{irpo,\pi}$ is defined as $s \approx_{irpo,\pi} t$ iff $\pi(\phi(s)) \approx_{rpo} \pi(\phi(t))$. The union of $\succ_{irpo,\pi}$ and $\approx_{irpo,\pi}$ is denoted as $\succsim_{irpo,\pi}$.*

The next proposition follows directly from Proposition 1 and Definition 7.

Proposition 2. *$\succ_{irpo,\pi}$ is a well-founded ordering compatible with the equivalence relation $\approx_{irpo,\pi}$.*

Now, we prove the IP-monotonicity of $\succsim_{irpo,\pi}$.

Theorem 7. *Let $\mathcal{R}$ be a non-overlaying TRS. If $l\sigma \succsim_{irpo,\pi} r\sigma$, for every rule $l \to r \in \mathcal{R}$ and substitution σ s.t. $l\sigma$ is argument normalized then $\succsim_{irpo,\pi}$ is IP-monotonic w.r.t. $\mathcal{R}$.*

Proof. We need to show that $s \twoheadrightarrow\!\!\!\!\!\!\shortparallel\ _i t$ implies $s \succsim_{irpo,\pi} t$, and we prove it by induction on the size of s. If this rewrite step is at position λ the result trivially follows. Otherwise, $s = f(\bar{s})$, $t = f(\bar{t})$ and for all $k = 1 \ldots |\bar{s}|$, either s_k is a normal form and $t_k = s_k$, or $s_k \twoheadrightarrow\!\!\!\!\!\!\shortparallel\ _i t_k$ and by induction hypothesis $s_k \succsim_{irpo,\pi} t_k$. Now, when $f \notin \mathcal{F}_{\mathcal{S}et}$, if $\pi(f)$ is either the empty set or a natural number the result is trivial; otherwise $s \succsim_{irpo,\pi} t$ is obtained using monotonicity and transitivity of $\succsim_{rpo}$. In case of $f \in \mathcal{F}_{\mathcal{S}et}$, first note that, by non-overlayingness, if $s_i \equiv s_j$, then $t_i \equiv t_j$. Therefore, if for some t_i, all the t_j's with $j < i$ are different from t_i (and hence t_i is not removed by the transformation ϕ) then, all the s_j's with $j < i$ are different from s_i. As consequence, to every element in $\phi(\bar{t})$ we can associate a distinct element in $\phi(\bar{s})$ that is greater w.r.t. $\succsim_{irpo,\pi}$, and hence, $s \succsim_{irpo,\pi} t$. $\square$

Combining Theorems 4, 7 and Proposition 5 we have that, for non-overlaying TRSs, the compatible pair $(\succsim_{irpo,\pi}, \succ_{irpo,\pi})$ can be effectively used for innermost termination proofs with DP.

Corollary 1. *A non-overlaying TRS $\mathcal{R}$ is innermost terminating if*

- $l\sigma \succsim_{irpo,\pi} r\sigma$ *for all* $l \rightarrow r \in \mathcal{R}$ *and substitution* σ *s.t.* $l\sigma$ *is argument normalized and*
- $s\sigma \succ_{irpo,\pi} t\sigma$ *for all* $\langle s,t\rangle \in \mathcal{DP}(\mathcal{R})$ *and substitution* σ *s.t.* $s\sigma$ *is argument normalized.*

Finally we point out that Algorithm 1 for the iRPO stability problem can be easily adapted for checking if $l\sigma \succsim_{irpo,\pi} r\sigma$, for every substitution σ s.t. $l\sigma$ is argument normalized.

5 Conclusions

In this paper we introduce the first syntactical ordering which can be used for proving innermost termination of non-terminating TRSs. The ordering is a variant of the most popular reduction ordering, RPO, and we call it the *innermost RPO*. The iRPO was obtained by considering, for some function symbols, sets instead of multisets of arguments. Hence, it is specially recommended for dealing with duplicated arguments in right-hand sides. The use of sets entails non-stability as drawback. However, for the (quasi-) orderings presented here, the problem of checking stability is decidable and co-NP-complete. The algorithm for doing this checking considers those m.g.u. which duplicate arguments in left-hand sides. But usually there are not many of such arguments. Therefore, we think that in many practical situations the stability of iRPO can be computed efficiently.

The iRPO enjoys a property, called IP-monotonicity, which is essential for innermost termination. This property demands monotonicity just after each (maximal) parallel innermost rewriting step. We believe that this weaker condition might be useful for defining other innermost-specific orderings.

References

1. T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1-2):133–178, 2000.
2. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
3. A. Ben-Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, (9):137–160, 1987.
4. C. Borralleras. *Ordering-based methods for proving termination automatically.* PhD thesis, Dpto. LSI, Universitat Politècnica de Catalunya, España, 2003.
5. C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. In *Proc. CADE*, volume 1831 of *LNAI*, pages 346–364, 2000.
6. N. Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.
7. M.L. Fernández. Relaxing monotonicity for innermost termination. *Information Processing Letters*, 93(3):117–123, 2005.
8. O. Fissore, I. Gnaedig, and H. Kirchner. Induction for innermost and outermost ground termination. Technical Report A01-R-178, LORIA, Nancy, France, 2001.
9. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Improving dependency pairs. In *Proc. LPAR*, volume 2850 of *LNAI*, pages 165–179, 2003.
10. B. Gramlich. On proving termination by innermost termination. In *Proc. RTA*, volume 1103 of *LNCS*, pages 93–107, 1996.
11. Nao Hirokawa and Aart Middeldorp. Polynomial interpretations with negative coefficients. In *Proc. AISC*, volume 3249 of *LNAI*, pages 185–198, 2004.
12. S. Kamin and J. J. Levy. Two generalizations of the recursive path ordering. Dept. of Computer Science, Univ. of Illinois, Urbana, IL, 1980.
13. K. Kusakari, M. Nakamura, and Y. Toyama. Argument filtering transformation. In *Proc. PPDP*, volume 1702 of *LNCS*, pages 47–61, 1999.
14. C. S. Lee, N. D. Jones, and A. M. Ben-Amram. The size-change principle for program termination. In *Proc. POPL*, pages 81–92, 2001.
15. S. Lucas. Polynomials for proving termination of context-sensitive rewriting. In *Proc. FOSSACS*, volume 2987 of *LNCS*, pages 318–332, 2004.
16. Z. Manna and S. Ness. On the termination of Markov algorithms. In *Proc. Int. Conf. on System Science*, pages 789–792, 1970.
17. M. J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *LNCS*. 1977.
18. M.R.K. Krishna Rao. Some characteristic of strong innermost normalization. *Theoretical Computer Science*, (239):141–164, 2000.
19. J. Steinbach and H. Xi. Freezing - termination proofs for classical, context-sensitive and innermost rewriting. Technical report, Institut für Informatik, T.U. München, Germany, 1998.
20. R. Thiemann and J. Giesl. Size-change termination for term rewriting. In *Proc. RTA*, volume 2706 of *LNCS*, pages 264–278, 2003.
21. Y. Toyama. Counterexamples to termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25:141–143, 1987.