Doctoral Dissertation

Unifying Semantic Path Order and Weighted Path Order

Teppei Saito

Supervisor: Nao Hirokawa

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Information science]

March 2026

Abstract

Orders play important roles in analysis of term rewrite systems. Among others, semantic path orders by Kamin and Lévy and weighted path orders by Yamada, Kusakari and Sakabe are known to be quite powerful. The thesis presents a generalization of these two classes of orders, which is useful for not only termination analysis but also for reachability analysis.

keywords: *term rewriting, termination, reachability, semantic path order, weighted path order*

Acknowledgments

First of all, I am grateful to Nao Hirokawa-sensei (my sensei) for introducing me to the world of term rewriting. Without his continuous support, this thesis would not be present. I owe a lot of knowledge, notations, hikes, sake and wine to Aart Middeldorp-sensei (the sensei of my sensei). Thank you too, Noriko-san and Toki-san, for the nice dinners and taking me to the Krampus parade, and Rika-san for the nice T-shirts. Among them, the ARI one is my favorite. I thank Akihisa Yamada-sensei for the valuable feedback on this thesis, and the stimulating discussions about the weighted path order and more. I also thank René Thiemann for the detailed comments and interesting questions about the thesis. In addition, when I was in Innsbruck for half a year, he helped me a lot, in particular about formalization of the semantic path order. I still remember his story about meeting my sensei for the first time, at their very first conference! In Innsbruck I was also helped by many kind people in the computational logic group: Gernot Baumgartner, Martina Ingenhaeff, Cezary Kaliszyk, Dohan Kim, Christina Kirk, Fabian Mitterwallner, Johannes Niederhauser, and Jonas Schöpf. I must thank for other TRS people: Takahito Aoto-sensei and Naoki Nishida-sensei were nice colleagues in the ARI project. Munehiro Iwami-sensei gave me an opportunity to think about use of the semantic path order for studying combinators. Jürgen Giesl and Jan-Christoph Kassing asked me valuable questions whenever I presented at a conference or workshop. Actually, Chapter 4 and Section 3.3 of this thesis are answers to Jürgen's questions at the 19th WST and the 14th FroCoS, respectively. Vincent van Oostrom indirectly taught me Buchholz's method via my sensei and encouraged me at every occasion. Although I have never met him in person, I also thank Alfons Geser for his inspiring work and help with the literature. Perhaps it was not a coincidence that my hatsuyume of 2025 was about him. From the theorem proving community, Geoff Sutcliffe helped me with StarExec Miami for running the Confluence Competition, encouraged me to write a prover, and gave me the cute T-shirts. In addition, the email from Geoff lead me to Marktoberdorf Summer School 2025, where Jasmin Blanchette gave me the valuable comments on the generalized weighted path order. There I had the very Bavarian Sunday with Nils Lommen and Johannes Niederhauser. Tsubasa Takagi-san gave me the very practical advices about the funding application. I also thank my colleagues: Jiani Dai-san, Ziyu Guo-san, Hiroka Hondo-san, Fuyuki Kawano-san, Hanyuan Li-san, Kiraku Shintani-san, Yuta Sunagawa-san, Teppei Tanaka-san, and Wataru Yachi-san. Last but not least, I thank my parents and Risa for the moral support.

Contents

Chapter 1

Introduction

This chapter motivates the theory of semantic path orders and weighted path orders, and clarifies the contributions of the thesis.

1.1 Order-Based Techniques in Term Rewriting

Term rewrite systems (TRSs) are a model of symbolic computation. To build intuition, let us consider the following TRS modeling addition of natural numbers.

$$\mathsf{add}(0, y) \xrightarrow{1} y \qquad\qquad \mathsf{add}(\mathsf{s}(x), y) \xrightarrow{2} \mathsf{s}(\mathsf{add}(x, y))$$

Here, numbers are represented as *terms* built from the function symbols 0 and s: 0 represents zero, $\mathsf{s}(0)$ one, $\mathsf{s}(\mathsf{s}(0))$ two, and so on. The binary function symbol add corresponds to addition, and the specification is described by the two rules: rule 1 states that adding zero makes no difference, and rule 2 states that adding $x + 1$ to y amounts to incrementing (by one) the result of adding x to y. Indeed, $2 + 1 = 3$ can be computed by applying the rules to the corresponding term $\mathsf{add}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(0))$ until the rules are no longer applicable.

$$\underline{\mathsf{add}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(0))} \xrightarrow{2} \mathsf{s}(\underline{\mathsf{add}(\mathsf{s}(0), \mathsf{s}(0))}) \xrightarrow{2} \mathsf{s}(\mathsf{s}(\underline{\mathsf{add}(0, \mathsf{s}(0))})) \xrightarrow{1} \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))$$

This two-rule TRS can be regarded as a way to teach computers how to add numbers. More generally, TRSs provide a foundation for the area of automated deduction, which is concerned about automation of mathematical reasoning with computers.

The TRS of addition is *terminating*, which means that there is no infinite rewrite sequence $t_0 \to t_1 \to t_2 \to \cdots$ of terms. This property is relevant, for example, when we mechanize mathematical reasoning by means of TRSs. In our example, the termination of the TRS of addition guarantees that the corresponding rewriting-based mechanization of addition always terminates. However, because rules in TRSs are arbitrary in general, the property of termination cannot be taken for granted. For instance, if we extend the TRS with the commutativity rule $\mathsf{add}(x, y) \to \mathsf{add}(y, x)$, the property of termination no longer holds,

as witnessed by the infinite rewrite sequence

$$\mathsf{add}(0,\mathsf{s}(0)) \to \mathsf{add}(\mathsf{s}(0),0) \to \mathsf{add}(0,\mathsf{s}(0)) \to \cdots .$$

As in termination proving in general, termination is concluded by finding a suitable termination measure that decreases along each step of rewriting. For this purpose, so-called *reduction orders* give a convenient characterization of termination of TRSs. For our example of addition, termination of the TRS is equivalent to existence of a reduction order $>$ that orients all the rules:

$$\mathsf{add}(0,y) > y \qquad\qquad \mathsf{add}(\mathsf{s}(x),y) > \mathsf{s}(\mathsf{add}(x,y))$$

The polynomial interpretation method due to Lankford [69] is a way to construct such a reduction order. Roughly speaking, the method associates each function symbol with a polynomial, and thereby interprets terms into natural numbers. For our purpose, let us interpret the symbol 0 as 1, $\mathsf{s}(x)$ as $x+1$, and $\mathsf{add}(x,y)$ as $2x+y$. This interpretation $\mathcal{A}$ induces the reduction order $>_{\mathcal{A}}$ satisfying

$$\mathsf{add}(0,y) >_{\mathcal{A}} y \qquad\qquad \mathsf{add}(\mathsf{s}(x),y) >_{\mathcal{A}} \mathsf{s}(\mathsf{add}(x,y))$$

so we conclude the termination of the TRS. To give a taste of it, let us explain the second inequality: the left-hand side $\mathsf{add}(\mathsf{s}(x),y)$ is interpreted as $2(x+1)+y = 2x+y+2$, while the right-hand side $\mathsf{s}(\mathsf{add}(x,y))$ is interpreted as $2x+y+1$. That $2x+y+2 > 2x+y+1$ for all natural numbers x and y entails the second inequality.

The present work is concerned with order-based techniques (such as reduction orders) in the area of term rewriting (the study of TRSs). Now, let us briefly review existing work.

Reduction orders. As we have seen in the introductory example, reduction orders are a basic method for analyzing termination of TRSs. Formally, a TRS $\mathcal{R}$ is terminating if and only if there is a reduction order $>$ that orients all rules in $\mathcal{R}$, that is, $\mathcal{R} \subseteq >$. There has been a long line of research about reduction orders. Lankford's polynomial interpretation method [69] is an example of sematic methods to construct reduction orders. This method is now understood as a special case of (strictly) monotone algebras due to Zantema [120]. We also have syntactic methods for constructing reduction orders such as Dershowitz's recursive path order [32] (also called the multiset path order nowadays) and the lexicographic path order due to Kamin and Lévy [59]. These methods are parameterized by a quasi-order on function symbols called precedence. Moreover, it is also possible and effective to combine both of syntactic and semantic methods. Such examples are the Knuth–Bendix order (KBO) [63, 67, 75, 78, 111], the *semantic path order* (SPO) [26, 59], and the *weighted path order* (WPO) [117].

Reduction orders have been applied not only for termination analysis but also theorem proving procedures such as Knuth–Bendix completion [63], ordered completion [10], and superposition calculus [9]. Indeed, there is a recent trend to build orders for those methods [13, 14, 17, 20] motivated by higher-order superposition calculus [15, 18].

Reduction pairs. The dependency pair method [3, 47, 54, 55] is one of the most successful methods for automated termination analysis of TRSs. One of the strengths of the method is that it allows us to use *reduction pair*, which is a more general form of reduction order. As the name suggests, reduction pairs are pairs of orders, rather than single orders. In its simplest form, the dependency pair method states that a TRS $\mathcal{R}$ is terminating if and only if there is a reduction pair $(\geqslant, >)$ with $\mathcal{R} \subseteq \geqslant$ and $\mathsf{DP}(\mathcal{R}) \subseteq >$, where $\mathsf{DP}(\mathcal{R})$ consists of auxiliary rules called dependency pairs. In fact, the dependency pair method with reduction pairs is the core of powerful automatic termination analyzers for TRSs, such as AProVE [49], NaTT [116] and $\mathsf{T_TT_2}$ [66].

Co-rewrite pairs. The reachability problem in term rewriting asks whether a term s can be rewritten to another term t under a TRS $\mathcal{R}$, after suitably instantiating variables in s and t. Alternatively, it can be understood as a sort of safety problem about whether a good state s can reach a bad state t in the system. The problem and its variants appear as subproblems in, among others, the dependency pair method and also analysis of conditional TRSs (see for example [74, 94]). Due to its importance, since 2019 there has been a competition category for a variant of the reachability problem in the Confluence Competition [79]. *Co-rewrite pair* is a generalization of reduction pairs which characterizes the reachability problem [114]: s cannot be rewritten to t under $\mathcal{R}$ (even after instantiation) if and only if there is a co-rewrite pair $(\geqslant, >)$ such that $\mathcal{R} \subseteq \geqslant$ and $t > s$. The termination prover NaTT also features reachability analysis by co-rewrite pairs, and has stood on the podium in the competition many times.

Lexicographic path order. Now, let us explain Kamin and Lévy's lexicographic path order (LPO) [59], because this is useful to develop technical intuition about the main subjects of the thesis. As mentioned earlier, the reduction order is parameterized by a precedence, which is a quasi-order on function symbols. The introductory example of addition can be shown terminating also with this method: the LPO $>_{\mathsf{L}}$ induced by a precedence $\succeq$ with $\mathsf{add} \succ \mathsf{s}$ satisfies

$$\mathsf{add}(0, y) >_{\mathsf{L}} y \qquad\qquad \mathsf{add}(\mathsf{s}(x), y) >_{\mathsf{L}} \mathsf{s}(\mathsf{add}(x, y))$$

so the TRS is again proven terminating. The reasoning behind these inequalities is as follows. The first inequality holds because y occurs in $\mathsf{add}(0, y)$ as a

proper subterm. The second inequality is in fact more intricate: First the LPO compares the head symbols add with s. Since add $\succ$ s with respect to the precedence $\succeq$, it proceeds to recursive comparison with the argument of s, namely $\mathsf{add}(\mathsf{s}(x), y) >_{\mathsf{L}} \mathsf{add}(x, y)$. This inequality also holds, because the first argument of add is decreasing, i.e., $\mathsf{s}(x) >_{\mathsf{L}} x$. Our informal explanation indicates that, as opposed to Lankford's polynomial interpretation, the LPO is quite syntactic.

Semantic path order. One of the two subjects of this thesis is the semantic path order (SPO), which is a generalization of the lexicographic path order also due to Kamin and Lévy [59]. (This relation is indicated as the arrow from the LPO to the SPO in Fig. 1.1.) More precisely, the SPO is obtained from the LPO by generalizing precedence to relation on terms. This generalization allows us, for example, to use a polynomial interpretation instead of a precedence, enabling a combination of the LPO and the interpretation method. Although such a generalization does not yield a reduction order in general, Borralleras, Ferreira and Rubio have developed a reduction order version called *monotonic semantic path order* (MSPO), and thus obtained a powerful yet automatable method of termination analysis of TRSs [22, 25, 26]. The reduction order is so powerful that it can simulate a part of the dependency pair method, including the dependency graph technique (see [22, Chapter 7]). On the other hand, use of the semantic path order as *reduction pair* within the dependency pair method is not thoroughly investigated.

Weighted path order. The other subject of the thesis is the weighted path order (WPO) due to Yamada, Kusakari and Sakabe [117], which is another generalization of the lexicographic path order. The order takes two parameters: one is a precedence and the other is an algebra (such as a polynomial interpretation). Like the SPO, the WPO can be regarded as a merger of the LPO and the polynomial interpretation, and also generalizes the Knuth–Bendix order (see Fig. 1.1). Moreover, the WPO has dedicated versions as a reduction pair and as a co-rewrite pair, so it can be used more effectively in the dependency pair method and in non-reachability analysis [114], respectively. In fact, the WPO in combination with the dependency pair method has been implemented in the powerful automatic termination analyzers AProVE [60], NaTT [116] and T$_{\text{T}}$T$_2$ [89].

1.2 Overview and Contributions

So, currently we have two powerful yet seemingly different generalizations: the semantic path order and the weighted path order. Under such a situation, it is natural to ask whether it is possible to unify them. We answer to this question affirmatively. This is done by the very simple idea of the semantic path order

to generalize precedence to relation on terms: we generalize the weighted path order into one that uses relations on terms instead of algebra and precedence. In fact, the situation about the thus-obtained order, dubbed *generalized weighted path order* (GWPO), is parallel to that of the semantic path order: The generalized weighted path order is not a reduction order in general, but there is a reduction order version of it, which we call *monotonic generalized weighted path order* (MGWPO).

As is demonstrated by the WPO, having a graceful unifying framework for term orders tells us strictly more than having different methods separately, since it typically suggests stronger constructions of orders. This is also the case for our (M)GWPO. We demonstrate this with termination and reachability analysis of TRSs. Another advantage of unifying the semantic path order and the weighted path order is that it allows us to construct something in-between. This is in need, for example, if we are interested in ground totality (a relevant property for theorem proving), which the WPO can hold (under natural assumptions) while the MSPO cannot (even under natural assumptions).

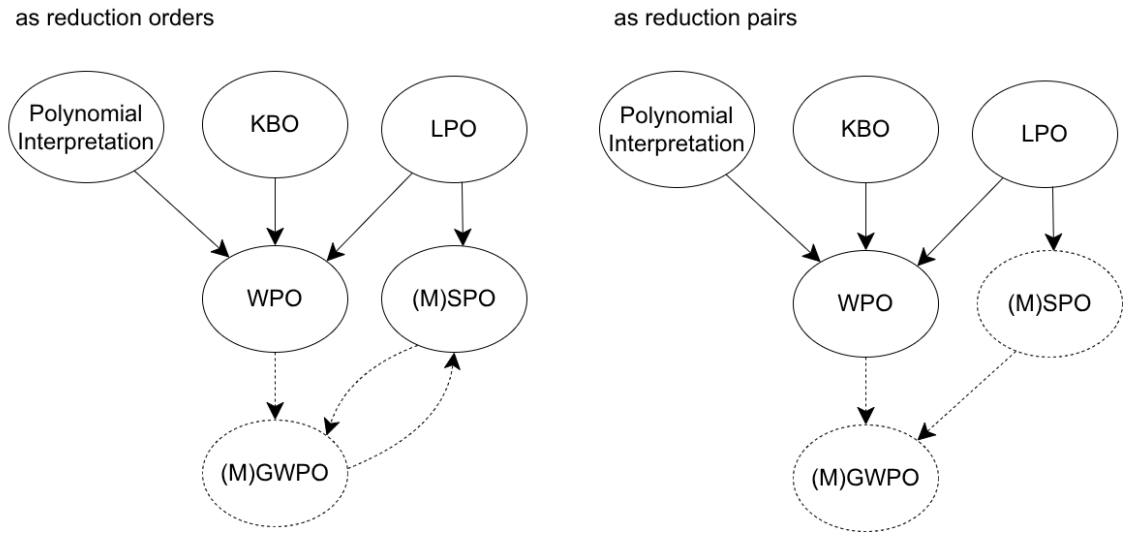

Figure 1.1: Relationship between orders. Dashed are our results.

Structure of Thesis

Chapter 2 is devoted to the preliminaries. The highlight is the section for the semantic path order, where a variant of the order necessary for this thesis is introduced. Chapter 3 introduces the monotonic generalized weighted path order as reduction order. Interestingly, it turns out that the (monotonic) generalized weighted path order can be simulated by the (monotonic) semantic path order

with some overhead, see Fig. 1.1. We also extend these results to rewriting modulo associativity and commutativity, and construct a ground-total extension of the weighted path order. Chapter 4 considers the reduction pair versions of the generalized weighted path order, which extend those of the weighted path order. As special cases of the GWPO, the reduction pair versions of the semantic path order are obtained for the first time. In contrast to the previous chapter, the simulation result by the semantic path order no longer holds. This is why there is no arrow from (M)GWPO to (M)SPO in Fig. 1.1. Chapter 5 studies the generalized weighted path order as co-rewrite pair. We begin with a simpler version (known as rewrite pair), and then consider a more advanced version. Chapter 6 concludes the thesis by discussing future and related work. Here, applications in the area of theorem proving are hinted.

Summary of Contributions

All in all, the contributions of the thesis are summarized as follows:

1. *Modest refinements for the theory of SPO* (Section 2.5). Although the theory of SPO [22, 26, 59] is not our invention, some results (namely, Theorem 2.5.10 and Proposition 2.5.15) necessary for relating it to the (G)WPO are missing in the existing literature. So, based on existing results, we introduce necessary variants and note some pitfalls (e.g., boundedness) often overlooked.

2. *(M)GWPO as reduction order* (Chapter 3). Although the (M)GWPO generalizes the WPO and the (M)SPO by definition, it turns out that the (M)SPO can also simulate the (M)GWPO. Such a result has been unknown, even for the original WPO. (This result is a sophistication of Geser's pioneering result that a generalized version of the KBO can be simulated by the SPO [42].) We further extend the result to rewriting modulo associativity and commutativity. In practice, the simulation result has two implications: the SPO allows us to bypass the weak simplicity requirement of the (G)WPO; conversely, the GWPO provides an optimized implementation of the SPO which avoids some of recursive comparison under a certain condition. Furthermore, in contrast to the SPO, we demonstrate that the GWPO is more suited for constructing a reduction order that goes beyond the WPO but still enjoys ground totality, which is a relevant property for theorem proving. In addition to these theoretical results, with a prototype implementation, we evaluate the methods on the standard benchmark problems (TPDB [98]) for termination analysis of TRSs.

3. *GWPO and SPO as reduction pair* (Chapter 4). There are two versions of the WPO as reduction pairs: a simple version and another with refinements. We extend these to the GWPO, and thus obtain dedicated reduction-pair

versions of the SPO for the first time ever (see Table 1.1). In particular, the refined version of the GWPO enjoys a nice property of simulating every non-trivial reduction pair. This is slightly stronger than that of the WPO only simulating non-trivial and *normal* reduction pairs, which do not fully cover, for example, the GWPO as reduction pair. Then we reveal that the simulation result between SPOs and GWPOs in the reduction order setting no longer holds, which gives a justification of having the GWPO in addition to the SPO. With a prototype implementation, we show that the GWPO is more powerful than the original WPO as reduction pair in TPDB.

4. *GWPO and SPO as co-rewrite pair* (Chapter 5). Based on the simple version of the GWPO as reduction pair, we develop two versions of the GWPO as co-rewrite pair and thus obtain corresponding SPO versions. These techniques allow us to solve reachability problems of term rewriting beyond the capability of existing automatic tools. We also demonstrate it with a prototype implementation.

Full experimental data and the prototype implementations are available at `https://data.non-terminat.ing/thesis-material.tar.gz`.

Finally, we add about the relationship of the present thesis to the existing publications where the author of the thesis is involved. Basically, Section 2.5 and Chapter 3 are a substantial extension of the conference paper [87], where a special form of the simulation result between the original WPO and the SPO is presented and a precursor of the prototype implementation is developed. We warn that their GWPO in the paper refers to a particular construction of our GWPO in the present thesis. (This is discussed in detail in Chapter 3). The prototype tools of Chapter 4 and Chapter 5 are based on those of [88] and [61], respectively. Other results are presented for the first time.

Table 1.1: First appearance of the methods.

	WPO	**GWPO**	**SPO**
reduction order	[117]	Chapter 3	[26, 59]
reduction pair	[117]	Chapter 4	Chapter 4
co-rewrite pair	[114]	Chapter 5	Chapter 5

Chapter 2

Preliminaries

This chapter provides the preliminary material of this thesis. More precisely, Sections 2.1 to 2.3 collect basic facts and notations used throughout the thesis. Sections 2.4 and 2.5 form a basis for Chapter 3; Section 2.6 is used in Section 3.4; Section 2.7 is used in Chapter 4; and Section 2.8 is used in Chapter 5. For a more detailed account of term rewriting, see the standard textbooks [8, 97].

2.1 Basic Order Theory

Let X and Y be sets. We write $X \subseteq Y$ if X is a subset of Y (so possibly $X = Y$), and $X \subsetneq Y$ if in addition $X \neq Y$. The empty set is denoted by $\varnothing$. The set of (ordered) pairs (x, y) from $x \in X$ and $y \in Y$ is denoted by $X \times Y$. For a natural number n, we write X^n for the set of all tuples $(x_1, \ldots, x_n)$ of length n, where $x_1, \ldots, x_n \in X$. As usual, we write X^* for the set of all tuples (of arbitrary but finite length) over X. The elements of X^* are also called *lists*.

A *binary relation* on a set X is a subset of $X \times X$. The empty relation is the empty set $\varnothing$. The relation $\{(x, x) \mid x \in X\}$ is the identity relation, denoted by $=$. Let $\to$ be a binary relation on X. The relation $\to$ is *transitive* if $x \to z$ whenever $x \to y$ and $y \to z$; it is *symmetric* if $y \to x$ whenever $x \to y$; it is *antisymmetric* if $x = y$ whenever $x \to y$ and $y \to x$; it is *reflexive* if $x \to x$ for all $x \in X$; and it is *irreflexive* if $x \to x$ does not hold for any $x \in X$. The relation *extends* another relation $\rightsquigarrow$ on X if $x \to y$ whenever $x \rightsquigarrow y$. The *complement* $\not\to$ is the relation such that $x \not\to y$ if and only if $x \to y$ does not hold. The *composition* $\to \cdot \rightsquigarrow$ is the relation such that $x \to \cdot \rightsquigarrow y$ if and only if there is some $z \in X$ with $x \to z$ and $z \rightsquigarrow y$. The *transitive closure* of $\to$ is the smallest transitive extension of $\to$, denoted by $\to^+$. Similarly, the *reflexive closure* is defined as the smallest reflexive extension, and the *symmetric closure* is defined as the smallest symmetric extension. In particular, the *transitive and reflexive closure* is denoted by $\to^*$, and the *reflexive, symmetric and transitive closure* is denoted by $\leftrightarrow^*$. The *inverse relation* $\leftarrow$ is the relation such that $x \leftarrow y$ if and only if $y \to x$. In this way, we denote the inverse relation by reversing the symbol.

A relation is a *strict order* if it is transitive and irreflexive. A relation is a

quasi-order (or *preorder*) if it is transitive and reflexive. It is called *partial order* if in addition it is antisymmetric. The reflexive closure of a strict order is a quasi-order. Conversely, a quasi-order $\geqslant$ induces the *strict part* $>$ defined as the relation such that $x > y$ if and only if $x \geqslant y$ but not $x \leqslant y$. The strict part is a strict order. Similarly, the *equivalence part* $\equiv$ induced by $\geqslant$ is the intersection of $\geqslant$ and its inverse $\leqslant$. The relation $\equiv$ is a so-called *equivalence relation*, i.e., it is reflexive, transitive and symmetric.

A relation $\rightsquigarrow$ on a set X is *well-founded* if there is no infinite sequence $x_0 \rightsquigarrow x_1 \rightsquigarrow \cdots$. For example, the standard order $>$ on the set $\mathbb{N}$ of natural numbers is well-founded, while that on the set $\mathbb{Z}$ of integers is not. The well-founded relation $\rightsquigarrow$ provides the principle of *well-founded induction* (or just induction) with respect to $\rightsquigarrow$, meaning that for an arbitrary property P of elements in X, every element x enjoys P if and only if every element x whose successors y all enjoy P also enjoys P. Here, y is a successor of x if $x \rightsquigarrow y$. The principle can be written as the following logical formula:

$$\forall x\, P(x) \quad \Longleftrightarrow \quad \forall x\, (\forall y\, (x \rightsquigarrow y \Longrightarrow P(y)) \Longrightarrow P(x))$$

When we prove the left by proving the right instead, we say, for example, we perform (well-founded) induction on x with respect to $\rightsquigarrow$. The so-called mathematical induction on $\mathbb{N}$ is well-founded induction with respect to $\rightsquigarrow$, where $x \rightsquigarrow y$ is defined as $x = y + 1$. Using the standard order $>$ on $\mathbb{N}$ instead of $\rightsquigarrow$ results in a convenient proof principle (known as complete induction or strong induction) used throughout the thesis. So, we may not even mention what it is with respect to.

A strict order $>$ on a set X is called *well-founded order* if it is well-founded. It is called *well-order* if it is *total*, meaning that $x > y$, $x = y$ or $y > x$ for all $x, y \in X$. Similarly, a quasi-order $\geqslant$ on a set X is *well-founded* if its strict part is well-founded, and *almost total* if $x \geqslant y$ or $y \geqslant x$ for all $x, y \in X$. As usual, we say that x is *minimal* with respect to $>$ if there is no y with $x > y$.

A pair $(\geqslant, >)$ of relations on the same set X is *compatible* if $> \cdot \geqslant \subseteq >$ and $\geqslant \cdot > \subseteq >$. The pair $(\geqslant, >)$ is an *order pair* if it is compatible, $\geqslant$ is a quasi-order and $>$ is a strict order; it is *well-founded* if $>$ is well-founded; and it is *normal* if $> \subseteq \geqslant$. We note that there are some variations of definitions of order pair. For instance, what is called order pair in [114, Definition 1] corresponds to normal order pair in our terminology. We say $(\geqslant, >)$ is *trivial* if $\geqslant$ is $X \times X$. Note that such an order pair is doomed to be the one with $>$ identical to $\varnothing$: if there are $x > y$ then $x > y \geqslant x$ and thus $x > x$, a contradiction. An order pair $(\geqslant, >)$ is *total* if $>$ is. Similarly, it is *almost total* if $\geqslant$ is. Note that totality alone does not imply almost-totality, witnessed by the order pair $(=, >)$ with $x > y$ on $X = \{x, y\}$. Note that normality is missing here. Actually, a total order pair is almost total if and only if it is normal. So, the union $> \cup =$ coincides with $\geqslant$ for a total and normal order pair $(\geqslant, >)$.

From now on, we recall basic constructions of order pairs. We begin with *lexicographic combination* which provides a way to combine multiple order pairs on the same set X into a more complex one on X.

Definition 2.1.1. *Let $(\geqslant_A, >_A)$ and $(\geqslant_B, >_B)$ be pairs of relations on the same set X. The* lexicographic combination *$(\geqslant_{AB}, >_{AB})$ is the pair of relations on X defined as follows:*

- *$x \geqslant_{AB} y$ if $x >_A y$, or both $x \geqslant_A y$ and $x \geqslant_B y$.*
- *$x >_{AB} y$ if $x >_A y$, or both $x \geqslant_A y$ and $x >_B y$.*

Lexicographic combination behaves well, in the sense that it preserves order-pairedness and well-foundedness.

Proposition 2.1.2. *Let $(\geqslant_A, >_A)$ and $(\geqslant_B, >_B)$ be order pairs on the same set. Then $(\geqslant_{AB}, >_{AB})$ is again an order pair. If both $(\geqslant_A, >_A)$ and $(\geqslant_B, >_B)$ are well-founded, so is $(\geqslant_{AB}, >_{AB})$.*

The next construction is *lexicographic product*. In contrast to lexicographic combination, order pairs on possibly different sets are combined into a strict order on the set of tuples.

Definition 2.1.3. *Let $(\geqslant_1, >_1), \ldots, (\geqslant_n, >_n)$ be n order pairs on sets $A_1, \ldots, A_n$, respectively. The* lexicographic product *$(\geqslant_1, >_1) \otimes \cdots \otimes (\geqslant_n, >_n)$ is the strict order $>$ defined on $A_1 \times \cdots \times A_n$ as follows: $(a_1, \ldots, a_n) > (b_1, \ldots, b_n)$ if there exists an index $k \in \{1, \ldots, n\}$ such that $a_k >_k b_k$ and $a_j \geqslant_j b_j$ for all $1 \leqslant j < k$.*

While it is not necessary for the present thesis, we could also consider the quasi-order resulting from lexicographic product. Like lexicographic combination, the product operation preserves well-foundedness.

Proposition 2.1.4. *The lexicographic product of order pairs $(\geqslant_1, >_1), \ldots, (\geqslant_n, >_n)$ is well-founded if every $>_i$ is well-founded.*

The third construction is lexicographic extension, which turns an order pair into one on the set of lists.

Definition 2.1.5. *Let $(\geqslant, >)$ be a pair of relations on a set A. The* lexicographic extension *$(\geqslant^{\mathsf{lex}}, >^{\mathsf{lex}})$ are pairs of relations on A^* defined recursively: $(a_1, \ldots, a_m) \geqslant^{\mathsf{lex}} (b_1, \ldots, b_n)$ if one of the following cases holds.*

- *$n = 0$*
- *$m, n > 0$ and $a_1 > b_1$*
- *$m, n > 0$ and $a_1 \geqslant b_1$, and moreover $(a_2, \ldots, a_m) \geqslant^{\mathsf{lex}} (b_2, \ldots, b_n)$*

Similarly, $(a_1, \ldots, a_m) >^{\mathsf{lex}} (b_1, \ldots, b_n)$ *if one of the following cases holds.*

- $m > 0$ *and* $n = 0$
- $m, n > 0$ *and* $a_1 > b_1$
- $m, n > 0$ *and* $a_1 \geqslant b_1$*, and moreover* $(a_2, \ldots, a_m) >^{\mathsf{lex}} (b_2, \ldots, b_n)$

It is known that $(\geqslant^{\mathsf{lex}}, >^{\mathsf{lex}})$ is an order pair on A^* if $(\geqslant, >)$ is. However, $>^{\mathsf{lex}}$ is not well-founded even if $>$ is. This can be illustrated by lists of natural numbers:

$$(1) >^{\mathsf{lex}} (0,1) >^{\mathsf{lex}} (0,0,1) >^{\mathsf{lex}} \cdots$$

Well-foundedness is recovered by restricting A^* by a fixed length, see [93, Section 3] for a related discussion. Given a set A, we write $A^{\leqslant n}$ for the union of A^i for all natural numbers $i \leqslant n$.

Proposition 2.1.6. *If* $>$ *is well-founded, then the restriction of* $>^{\mathsf{lex}}$ *to* $A^{\leqslant n}$ *for a natural number* $n \in \mathbb{N}$ *is well-founded.*

Quite often, for a single strict order $>$ (rather than an order pair), we consider the strict part $>^{\mathsf{lex}}$ of the lexicographic extension of $(=, >)$, where $=$ is the identity relation. In that case, we may simply call $>^{\mathsf{lex}}$ the lexicographic extension of $>$.

The last construction is based on *multisets*, which are a variant of sets where multiplicity of elements counts. Formally, a multiset on a set X is a function $M : X \to \mathbb{N}$ with $M(x) > 0$ for only finitely many $x \in X$. (So what we consider here are finite multisets.) The set of all multisets on X is denoted by $\mathcal{M}(X)$. For example, the function $M : \mathbb{N} \to \mathbb{N}$ defined by $M(0) = 2$, $M(7) = 3$ and $M(x) = 0$ for other $x \in \mathbb{N}$ is a multiset on $\mathbb{N}$ which contains 2 zeros and 3 sevens. We write M as $\{0, 0, 7, 7, 7\}$ by abusing the notation for sets. Similarly, $x \in M$ means $M(x) > 0$, the empty multiset $\varnothing$ is the constant function that always returns 0, and for multisets M and N, we write $M \uplus N$ for the function defined by $(M \uplus N)(x) = M(x) + N(x)$. Now, suppose that X is endowed with an order pair $(\geqslant, >)$. The *multiset extension* $(\geqslant^{\mathsf{mul}}, >^{\mathsf{mul}})$ is the pairs of relations on $M(X)$ defined as follows: $M \geqslant^{\mathsf{mul}} N$ if M and N can be written in the form of $M = \{x_1, \ldots, x_n\} \uplus M'$ and $N = \{y_1, \ldots, y_n\} \uplus N'$ in such a way that $x_i \geqslant y_i$ for all i and moreover, for all $y \in N'$ there is some $x \in M'$ with $x > y$; if in addition $M' \neq \varnothing$, we write $M >^{\mathsf{mul}} N$. For example, we have $\{0, 3\} >^{\mathsf{mul}} \{0, 1, 2, 2\}$. If $(\geqslant, >)$ is a well-founded order pair, so is $(\geqslant^{\mathsf{mul}}, >^{\mathsf{mul}})$ [36, 102].

2.2 Term Rewriting

We begin with the definition of terms. A *signature* $\mathcal{F}$ is a designated set of function symbols. Each function symbol is associated with a unique natural

number called *arity*, which is regarded as the number of arguments of the function. Function symbols are called *constants* if their arity is 0, *unary* if their arity is 1, and *binary* if their arity is 2. When we need to indicate the arity n of a function symbol f, we write $f^{(n)}$ for f. Let $\mathcal{V}$ be a designated countably infinite set of *variables* with $\mathcal{F} \cap \mathcal{V} = \varnothing$. The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ of *terms* are defined inductively: every variable is a term, and $f(t_1, \ldots, t_n)$ is a term if $f^{(n)} \in \mathcal{F}$ and $t_1, \ldots, t_n$ are terms. The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ may be denoted by $\mathcal{T}$ when $\mathcal{F}$ and $\mathcal{V}$ are clear from the context. The *size* $|t|$ of a term t is the number of function symbols and variables occurring in t. Similarly, given a term t and a variable x, we write $|t|_x$ for the number of occurrences of x in t.

We define a few convenient notations for terms. For unary symbols f and natural numbers n, we may use *iteration* notation $f^n(t)$ defined inductively: $f^0(t) = t$ and $f^{n+1}(t) = f(f^n(t))$. Similarly, for binary function symbols, we may use *infix* notation for readability. For example, for a binary symbol $+$, we may write $t_1 + t_2$ for $+(t_1, t_2)$. Also, we may use vector notation $t = f(\bar{t})$ to indicate that t is a term headed by the function symbol f. In this case, f is the *root symbol* of t, denoted by $\mathrm{root}(t)$.

Let $\Box$ be a fresh constant called *hole* with $\Box \notin \mathcal{F}$. *Contexts* are terms over $\mathcal{F} \cup \{\Box\}$ that contain exactly one hole. The term resulting from replacing $\Box$ in a context C by a term t is denoted by $C[t]$. We write $s \trianglerighteq t$ if there is a context C with $s = C[t]$. In such a case, t is a *subterm* of s, or conversely, s is a *superterm* of t. The strict part of $\trianglerighteq$ is denoted by $\triangleright$. Similarly, $s \triangleright t$ reads as t is a *proper subterm* of s, or conversely as s is a *proper superterm* of t. A *substitution* is a mapping σ from variables to terms such that $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ is finite. The application $t\sigma$ of a substitution σ to a term t is inductively defined as follows: $t\sigma = \sigma(t)$ if t is a variable, and $t\sigma = f(t_1\sigma, \ldots, t_n\sigma)$ if $t = f(t_1, \ldots, t_n)$.

An ordered pair (ℓ, r) of terms is called *equation*, which we may denote by $\ell \approx r$. A set $\mathcal{E}$ of equations is called *equational system*. The relation $\to_{\mathcal{E}}$ is defined on terms as follows: $s \to_{\mathcal{E}} t$ if there exist an equation $\ell \approx r \in \mathcal{E}$, a context C, and a substitution σ such that $s = C[\ell\sigma]$ and $t = C[r\sigma]$ hold. When $C = \Box$, it is written as $s \xrightarrow{\epsilon}_{\mathcal{E}} t$.

An equation (ℓ, r) is said to be a *rewrite rule* if ℓ is not a variable and every variable in r occurs in ℓ. Rewrite rules (ℓ, r) are written by $\ell \to r$. A set of rewrite rules is called a *term rewrite system* (TRS). A TRS $\mathcal{R}$ is *non-duplicating* if every rule $\ell \to r \in \mathcal{R}$ is non-duplicating, meaning that $|\ell|_x \geqslant |r|_x$ for all variables x.

Example 2.2.1. *In Section 1.1 we have already seen the TRS $\mathcal{R}$ of addition.*

$$\mathsf{add}(0, y) \xrightarrow{1} y \qquad\qquad \mathsf{add}(\mathsf{s}(x), y) \xrightarrow{2} \mathsf{s}(\mathsf{add}(x, y))$$

The system is non-duplicating. A possible rewrite sequence is:

$$\mathsf{add}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(0)) \to_{\mathcal{R}} \mathsf{s}(\mathsf{add}(\mathsf{s}(0), \mathsf{s}(0))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{add}(0, \mathsf{s}(0)))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{s}(0)))$$

We use this system for illustrating basic techniques later.

Let $\hookrightarrow$ be a relation on terms. We say that

- $\hookrightarrow$ is *closed under contexts* or just *monotone* if $C[s] \hookrightarrow C[t]$ holds whenever $s \hookrightarrow t$ and C is a context;
- $\hookrightarrow$ is *closed under substitutions* or just *stable* if $s\sigma \hookrightarrow t\sigma$ holds whenever $s \hookrightarrow t$ and σ is a substitution;
- $\hookrightarrow$ is a *rewrite relation* if it is stable and monotone;
- $\hookrightarrow$ has the *subterm property* if $s \hookrightarrow t$ whenever $s \rhd t$; and
- $\hookrightarrow$ is *harmonious* to another relation $\rightsquigarrow$ on terms if

$$s_i \rightsquigarrow t \quad \Longrightarrow \quad f(s_1, \ldots, s_i, \ldots, s_n) \hookrightarrow f(s_1, \ldots, t, \ldots, s_n)$$

  for all n-ary function symbols f, integers $1 \leqslant i \leqslant n$, and terms $s_1, \ldots, s_n, t$.

It is not hard to see that $\hookrightarrow$ is closed under contexts if and only if $\hookrightarrow$ is harmonious to itself. We say that a pair $(\gtrsim, >)$ of relations on terms is *stable* if $\gtrsim$ and $>$ are closed under substitutions.

A term that contains no variables is called *ground*. A strict order $>$ on terms is *ground total* if either $s > t$ or $t > s$ holds for all distinct ground terms s, t.

A term t is *terminating* with respect to a relation $\hookrightarrow$ on terms if there is no infinite sequence $t \hookrightarrow t_1 \hookrightarrow t_2 \hookrightarrow \cdots$ starting from t. *Termination* of the relation $\hookrightarrow$ is defined as absence of non-terminating terms. A TRS $\mathcal{R}$ is *terminating* if $\to_{\mathcal{R}}$ is. Given TRSs $\mathcal{R}$ and $\mathcal{S}$, the relation $\to_{\mathcal{R}/\mathcal{S}}$ is defined on terms as follows: $s \to_{\mathcal{R}/\mathcal{S}} t$ if $s \to_{\mathcal{S}}^* u \to_{\mathcal{R}} v \to_{\mathcal{S}}^* t$ for some terms u and v. We say that $\mathcal{R}$ is *relatively terminating* with respect to $\mathcal{S}$, or $\mathcal{R}/\mathcal{S}$ is terminating, if $\to_{\mathcal{R}/\mathcal{S}}$ is terminating.

Termination of TRSs is often shown by orders. We say that a rewrite relation is a *rewrite preorder* or *reduction order* if it is a preorder or a well-founded order, respectively.

Proposition 2.2.2. *A TRS $\mathcal{R}$ is terminating if $\mathcal{R} \subseteq >$ for some reduction order $>$.*

A powerful method for automated termination analysis is the dependency pair method (to be explained in Section 2.7). Rather than reduction orders, this method is based on *reduction pairs*, namely well-founded stable order pairs $(\gtrsim, >)$ with $\gtrsim$ closed under contexts. A reduction pair $(\gtrsim, >)$ is *monotone* if $>$ is also closed under contexts. We note that $>$ is a reduction order for a monotone reduction pair $(\gtrsim, >)$. Conversely, given a reduction order $>$, the pair $(\gtrsim, >)$ of the reflexive closure $\gtrsim$ and $>$ is a monotone reduction pair.

2.3 Interpretation-Based Orders

Algebra (or interpretation) is useful for constructing reduction orders/pairs. Let $\mathcal{F}$ be a set of function symbols. An *$\mathcal{F}$-algebra* is a pair $(A, \{f_{\mathcal{A}}\}_{f\in\mathcal{F}})$, where A is a set called a *carrier*, and $f_{\mathcal{A}}$ is an n-ary function on A (called an *interpretation function*) associated with each $f^{(n)} \in \mathcal{F}$. Let $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f\in\mathcal{F}})$ be an algebra. A mapping from $\mathcal{V}$ to A is called an *assignment* for $\mathcal{A}$. The interpretation $[\alpha]_{\mathcal{A}}(t)$ of a term t under an assignment α is inductively defined as follows: $[\alpha]_{\mathcal{A}}(t) = \alpha(t)$ if t is a variable, and $[\alpha]_{\mathcal{A}}(t) = f_{\mathcal{A}}([\alpha]_{\mathcal{A}}(t_1), \ldots, [\alpha]_{\mathcal{A}}(t_n))$ if $t = f(t_1, \ldots, t_n)$. Assume that the carrier A is endowed with an order pair $(\geqslant, >)$. We write $s >_{\mathcal{A}} t$ if $[\alpha]_{\mathcal{A}}(s) > [\alpha]_{\mathcal{A}}(t)$ for all assignments α. The relation $>_{\mathcal{A}}$ is a stable strict order. Similarly we write $s \geqslant_{\mathcal{A}} t$ if $[\alpha]_{\mathcal{A}}(s) \geqslant [\alpha]_{\mathcal{A}}(t)$ holds for all assignments α. The relation $\geqslant_{\mathcal{A}}$ is a stable quasi-order, and satisfies compatibility with $>_{\mathcal{A}}$, namely $\geqslant_{\mathcal{A}} \cdot >_{\mathcal{A}} \subseteq >_{\mathcal{A}}$ and $>_{\mathcal{A}} \cdot \geqslant_{\mathcal{A}} \subseteq >_{\mathcal{A}}$. An i-th argument position of $f^{(n)} \in \mathcal{F}$ with $1 \leqslant i \leqslant n$ is said to be

- *strictly simple* if $f_{\mathcal{A}}(a_1, \ldots, a_i, \ldots, a_n) > a_i$ for all $a_1, \ldots, a_n \in A$;
- *weakly simple* if $f_{\mathcal{A}}(a_1, \ldots, a_i, \ldots, a_n) \geqslant a_i$ for all $a_1, \ldots, a_n \in A$;
- *strictly monotone* if $f_{\mathcal{A}}(a_1, \ldots, a_i, \ldots, a_n) > f_{\mathcal{A}}(a_1, \ldots, b, \ldots, a_n)$ follows from $a_i > b$ for all $a_1, \ldots, a_n, b \in A$; and
- *weakly monotone* if $f_{\mathcal{A}}(a_1, \ldots, a_i, \ldots, a_n) \geqslant f_{\mathcal{A}}(a_1, \ldots, b, \ldots, a_n)$ follows from $a_i \geqslant b$ for all $a_1, \ldots, a_n, b \in A$.

The terminology above carries over to subsets $\mathcal{G} \subseteq \mathcal{F}$: For example, we say that $\mathcal{A}$ is strictly simple for $\mathcal{G}$ if every argument position of $f \in \mathcal{G}$ is so. Moreover, if $\mathcal{G} = \mathcal{F}$, we may simply call $\mathcal{A}$ strictly simple. The algebra $\mathcal{A}$ is *well-founded* if $>$ is well-founded; *normal* if $> \subseteq \geqslant$; and *trivial* if the carrier A is a singleton set. It is known that if $>$ is well-founded, so is $>_{\mathcal{A}}$. Similarly, if $\mathcal{A}$ is normal, so is $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$. If $\mathcal{A}$ is trivial, then $\geqslant_{\mathcal{A}}$ degenerates to $\mathcal{T} \times \mathcal{T}$, and $>_{\mathcal{A}}$ to $\varnothing$. If $\mathcal{A}$ is a weakly monotone algebra, $\geqslant_{\mathcal{A}}$ is a rewrite preorder. If $\mathcal{A}$ is a strictly monotone algebra, $>_{\mathcal{A}}$ is a reduction order. If $\mathcal{A}$ is strictly (resp. weakly) simple, $>_{\mathcal{A}}$ (resp. $\geqslant_{\mathcal{A}}$) has the subterm property. An important consequence is:

Proposition 2.3.1 ([120]). *If $\mathcal{A}$ is a weakly monotone and well-founded algebra, then $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ is a reduction pair. If in addition $\mathcal{A}$ is strictly monotone, then $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ is a monotone reduction pair.*

Lankford's polynomial interpretation [69] is a popular choice for algebras. Formally, a *polynomial interpretation* over a set A of integers is an algebra whose carrier is A (ordered as usual) and interpretations of function symbols are given by polynomials over integers. Normally we use $\mathbb{N}$ for the carrier A but also the

set $\mathbb{Z}_{\leqslant 0}$ of negative integers (including zero) when reachability is concerned. A *linear polynomial interpretation* $\mathcal{A}$ is a polynomial interpretation such that the interpretation $f_{\mathcal{A}}(x_1, \ldots, x_n)$ is of the form $c_0 + c_1x_1 + \cdots + c_nx_n$ for each function symbol $f^{(n)}$. Suppose that $\mathcal{A}$ is a linear polynomial interpretation over $\mathbb{N}$. In this case, as $f_{\mathcal{A}}(x_1, \ldots, x_n) = c_0 + c_1x_1 + \cdots + c_nx_n$ are well-defined mappings from $\mathbb{N}^n$ to $\mathbb{N}$, automatically $c_0, c_1, \ldots, c_n \in \mathbb{N}$ holds. So the interpretation is not only well-founded and normal but also weakly monotone. Moreover, the i-th argument of a function symbol f is strictly monotone, and strictly and weakly simple if and only if $c_i > 0$.

The next proposition is convenient for mechanizing $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ for a linear polynomial interpretation $\mathcal{A}$, as it reduces the task to comparison of coefficients. For functions $f, g : \mathbb{N}^n \to \mathbb{N}$, we write $f \geqslant g$ if $f(x_1, \ldots, x_n) \geqslant g(x_1, \ldots, x_n)$ for all $x_1, \ldots, x_n \in \mathbb{N}$ and similarly, we write $f > g$ on $\mathbb{N}$ if $f(x_1, \ldots, x_n) > g(x_1, \ldots, x_n)$ for all $x_1, \ldots, x_n \in \mathbb{N}$.

Proposition 2.3.2. *Let $f(x_1, \ldots, x_n) = c_0 + c_1x_1 + \cdots + c_nx_n$ and $g(x_1, \ldots, x_n) = d_0 + d_1x_1 + \cdots + d_nx_n$ be mappings from $\mathbb{N}^n$ to $\mathbb{N}$. Then the next statements hold.*

- *$f \geqslant g$ if and only if $c_0 \geqslant d_0$ and $c_i \geqslant d_i$ for all $1 \leqslant i \leqslant n$.*
- *$f > g$ if and only if $c_0 > d_0$ and $c_i \geqslant d_i$ for all $1 \leqslant i \leqslant n$.*

Now, it is nice exercise to re-do the termination proof of Section 1.1 formally. Alongside we also explain how to automate it with computers.

Example 2.3.3. *Consider the TRS of addition (Example 2.2.1). To show the termination with Proposition 2.2.2, it suffices to find a strictly monotone linear polynomial interpretation $\mathcal{A}$ over $\mathbb{N}$ with $\mathcal{R} \subseteq >_{\mathcal{A}}$. Let $\mathsf{0}_{\mathcal{A}} = a$, $\mathsf{s}_{\mathcal{A}}(x) = bx + c$ and $\mathsf{add}_{\mathcal{A}}(x, y) = dx + ey + f$. Here a, b, c, d, e and f are natural numbers to be determined. By calculation, one can see that the requirement $\mathcal{R} \subseteq >_{\mathcal{A}}$ is equivalent to*

$$ey + ad + f > y \qquad bdx + ey + cd + f > bdx + bey + bf + c$$

for all $x, y \in \mathbb{N}$. By Proposition 2.3.2, these inequalities boil down to the following constraints:

$$e \geqslant 1 \qquad ad + f > 0 \qquad bd \geqslant bd \qquad e \geqslant be \qquad cd + f > bf + c$$

Also, to guarantee strict monotonicity, we additionally demand that b, d, e are all positive. A solution of these constraints is the one used in the introduction, namely:

$$a = 1 \qquad b = 1 \qquad c = 1 \qquad d = 2 \qquad e = 1 \qquad f = 0$$

So, the termination of $\mathcal{R}$ is witnessed by this interpretation.

To find a solution of inequality constraints as above, we can consider using satisfiability modulo theory (SMT) solvers. Unfortunately, the problem is a variant of Hilbert's 10th problem which is shown undecidable by a straightforward adaptation of [80, Lemma 2.2].[1] So, there is no guarantee for SMT solvers to always find a solution. A simple but practical approach (suggested for example in [117]) is to restrict non-constant coefficients (b, d, e in the example) at most one in order to keep the resulting constraints within decidable linear arithmetic with if-then-else expressions (supported by many SMT solvers), at the expense of expressiveness. For example, with this restriction the constraint $cd + f > bf + c$ can be expressed as

```
(> (+ (ite (= d 1) c 0) f) (+ (ite (= b 1) f 0) c))
```

in SMT-LIB [12] using if-then-else (`ite`). The prototype implementations of this thesis follow this approach for linear polynomial interpretations. Another possible approach is to restrict coefficients as well as constants (a, c, f in the example) below a fixed upper bound so that the constraints stay within decidable bit-vector arithmetic (or propositional SAT) [40].

A *max/plus interpretation* $\mathcal{A}$ associates a combination of $+$ and max for each function symbol. For simplicity, we only consider $\mathbb{N}$ as the carrier, and the interpretations are restricted to the following form (adapted from [117, Definition 7]): For each $f^{(n)} \in \mathcal{F}$ its interpretation is of the form $f_{\mathcal{A}}(x_1, \ldots, x_n) = \max\{c_0, c_1 + c_1'x_1, \cdots, c_n + c_n'x_n\}$ where c_0 is a natural number, $c_1, \ldots, c_n \in \mathbb{Z}$ and $c_1', \ldots, c_n' \in \{0, 1\}$.[2] The i-th argument of a function symbol f is weakly simple if and only if $c_i \geqslant 0$ and $c_i' > 0$. Every max/plus interpretation $\mathcal{A}$ is weakly monotone, well-founded and normal.

Now, let us consider how to mechanize $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ for a max/plus interpretation $\mathcal{A}$. Given a term, the interpretation can be messaged into the form $\max\{g_1, \ldots, g_m\}$ where $g_1, \ldots, g_m$ are linear univariate polynomials $cx + d$ with $c \in \{0, 1\}$ and $d \in \mathbb{Z}$. Again, comparison of max/plus interpretation is reduced to that of coefficients using the following fact:

Proposition 2.3.4. *Let $G, H : \mathbb{N}^\ell \to \mathbb{N}$ be functions defined as*

$$G(x_1, \ldots, x_\ell) = \max\{g_1, \ldots, g_m\} \qquad H(x_1, \ldots, x_\ell) = \max\{h_1, \ldots, h_n\}$$

where m and n are positive integers, and $g_1, \ldots, g_m, h_1, \ldots, h_n$ can be written in the form $cx_k + d$ for some $c \in \{0, 1\}$, integer d and indices $1 \leqslant k \leqslant \ell$. Then:

- *$G \geqslant H$ if and only if for every $h_j = c_jx + d_j$ there exists $g_i = c_ix + d_i$ such that $c_i \geqslant c_j$ and $d_i \geqslant d_j$.*

[1] Even worse, the original problem (existence of a strictly monotone linear polynomial interpretation $\mathcal{A}$ over $\mathbb{N}$ with $\mathcal{R} \subseteq >_{\mathcal{A}}$ for a finite TRS $\mathcal{R}$) is already undecidable [80].

[2] Automation techniques for a more general form of interpretations are studied in [41].

- *$G > H$ if and only if for every $h_j = c_j x + d_j$ there exists $g_i = c_i x + d_i$ such that $c_i \geqslant c_j$ and $d_i > d_j$.*

Proof. We only show the first claim, as the second can be shown in a similar fashion. The if direction is trivial. To show the only-if direction, assume $G \geqslant H$ and let $h_j = c_j x_k + d_j$. If $c_j = 0$ then we choose $g_i = c_i x + d_i$ with the largest d_i. We can write $h_j = 0 \cdot x + d_j$, and also $d_i = G(0,\dots,0) \geqslant H(0,\dots,0) \geqslant d_j$ as desired. If $c_j = 1$, then from $G \geqslant H$ there is some $g_i = x_k + d_i$. Among them, we choose $g_i = x_k + d_i$ with the largest d_i. Now, by taking a sufficiently large natural number for x_k we have

$$G(0,\dots,x_k,\dots,0) = x_k + d_i \geqslant H(0,\dots,x_k,\dots,0) \geqslant x_k + d_j,$$

so $d_i \geqslant d_j$ as desired. ∎

For the only-if direction of Proposition 2.3.4, it is crucial that $c \in \{0,1\}$ for the inner polynomials $cx_k + d$: For example, let $G(x) = \max\{1, 2x\}$ and $H(x) = \max\{x\} = x$. Although $G > H$, the criterion does not apply.

Example 2.3.5. *Consider searching a max/plus interpretation $\mathcal{A}$*

$$\mathsf{f}_{\mathcal{A}}(x) = \max\{a, bx + c\} \qquad \mathsf{g}_{\mathcal{A}}(x) = \max\{d, ex + f\}$$

satisfying the constraint $\mathsf{f}(x) >_{\mathcal{A}} \mathsf{g}(\mathsf{f}(x))$. By calculation, the inequality boils down to

$$\mathsf{f}_{\mathcal{A}}(x) = \max\{a, bx + c\} > \max\{d, ae + f, bex + ce + f\} = \mathsf{g}_{\mathcal{A}}(\mathsf{f}_{\mathcal{A}}(x))$$

for all $x \in \mathbb{N}$. Proposition 2.3.4 transforms this to

$$\begin{gathered}(a > d \vee c > d) \wedge (a > ae + f \vee c > ae + f) \\ \wedge ((be = 0 \wedge a > ce + f) \vee (b \geqslant be \wedge c > ce + f))\end{gathered}$$

A possible solution satisfying the implicit requirements such as $b, e \in \{0,1\}$ is $a = d = 0$, $b = c = e = 1$ and $f = -1$. So, we obtain the corresponding interpretation $\mathsf{f}_{\mathcal{A}}(x) = x + 1$ and $\mathsf{g}_{\mathcal{A}}(x) = \max\{0, x - 1\}$.

Concerning automatic search of algebras, our prototype implementations use Proposition 2.3.4 to reduce the task to constraint solving in decidable linear arithmetic by an SMT solver. A key to enable this is the restriction $c'_1, \dots, c'_n \in \{0,1\}$ in our definition of max/plus interpretation, which corresponds to $b, e \in \{0,1\}$ in the example.

We say an algebra $\mathcal{A}$ is *almost total* if $(\geqslant, >)$ is almost total, and *total* if $(\geqslant, >)$ is total. Polynomial interpretations and max/plus interpretations are examples of total and normal (thus also almost total) algebras. Relations $\geqslant_{\mathcal{A}}$ are almost total on ground terms for almost total algebras $\mathcal{A}$. However, $>_{\mathcal{A}}$ are not total

on ground terms in general, even if $\mathcal{A}$ is normal. For example, consider the polynomial interpretation $\mathcal{A}$ with $\mathsf{a}_\mathcal{A} = \mathsf{b}_\mathcal{A} = 0$.

A *term algebra* is an algebra $\bar{\mathcal{T}}$ whose carrier is the set of terms and function symbols are interpreted as themselves, i.e., $f_{\bar{\mathcal{T}}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$ for all n-ary function symbols f and terms $t_1, \ldots, t_n$. Let $(\geqslant, >)$ be a reduction pair. Then term algebra $\bar{\mathcal{T}}$ equipped with $(\geqslant, >)$ is weakly monotone, and the induced order pair $(\geqslant_{\bar{\mathcal{T}}}, >_{\bar{\mathcal{T}}})$ is in fact identical to $(\geqslant, >)$. This fact allows us to switch from order pairs to algebras.

2.4 Weighted Path Order and its Special Cases

So-called path orders are a rather syntactical way to construct reduction orders (as opposed to the interpretation method). They typically use quasi-orders on the signature called *(quasi-)precedences*. A precedence that is a partial order is called *partial precedence*. Similarly, a precedence that is a total order is called *total precedence*. A quasi-precedence $\succeq$ is called *well-founded* if its strict part $\succ$ is well-founded. The *lexicographic path order* (LPO) [59] is a typical example of path orders that uses precedence.

Definition 2.4.1. *Let $\succeq$ be a precedence. The* lexicographic path order $>_\mathsf{L}$ *is defined on terms as follows: $s >_\mathsf{L} t$ if $s = f(s_1, \ldots, s_m)$ and one of the following conditions holds.*

1. *$s_i \geqslant_\mathsf{L} t$ for some $1 \leqslant i \leqslant m$.*
2. *$t = g(t_1, \ldots, t_n)$ and $s >_\mathsf{L} t_j$ for all $1 \leqslant j \leqslant n$, and moreover*
   a. *$f \succ g$, or*
   b. *$f \succeq g$ and $(s_1, \ldots, s_m) >_\mathsf{L}^\mathsf{lex} (t_1, \ldots, t_n)$.*

Here $\geqslant_\mathsf{L}$ is the reflexive closure of $>_\mathsf{L}$ and $>^\mathsf{lex}$ is the lexicographic extension of $>$.

As the name suggests, the LPO uses lexicographic extension to compare arguments. The version that uses multiset extension instead is known as the *multiset path order* [32], and the one that employs both is known as the *recursive path order* (with status). See also [103, Definition 4] for a most general version, and [91] for a historical account.

When the signature is infinite, lexicographic path orders are not always well-founded, even if the precedences are well-founded, see Proposition 2.1.6.

Example 2.4.2. *Consider the signature consisting of $\mathsf{a}^{(0)}$, $\mathsf{b}^{(0)}$, and $\mathsf{f}_i^{(i)}$ for all numbers $i \in \mathbb{N}$. Let $\succeq$ be a well-founded precedence satisfying $\mathsf{a} \succ \mathsf{b}$ and $\mathsf{f}_i \succeq \mathsf{f}_j$ for all $i, j \in \mathbb{N}$. The LPO $>_\mathsf{L}$ induced from $\succeq$ admits the infinite chain:*

$$\mathsf{f}_1(\mathsf{a}) >_\mathsf{L} \mathsf{f}_2(\mathsf{b}, \mathsf{a}) >_\mathsf{L} \mathsf{f}_3(\mathsf{b}, \mathsf{b}, \mathsf{a}) >_\mathsf{L} \cdots$$

See [104, Section 3] and [93, Section 3] for related discussions.

Well-foundedness of $>_{\mathsf{L}}$ is restored by assuming existence of an upper bound of arity, cf. Proposition 2.1.6. We refer to this property as *boundedness* of the signature. Needless to say, a signature is bounded whenever it is finite.

Proposition 2.4.3 ([59]). *Suppose that the signature is bounded. For every well-founded precedence, the induced lexicographic path order is a reduction order with the subterm property.*

For example, one can use the lexicographic path order with a precedence $\succeq$ with $\mathsf{add} \succ \mathsf{s}$ to show that the TRS of addition (Example 2.2.1) is terminating. More precisely, $\mathsf{add}(0, y) >_{\mathsf{L}} y$ is shown by case 1, and $\mathsf{add}(\mathsf{s}(x), y) >_{\mathsf{L}} \mathsf{s}(\mathsf{add}(x, y))$ is shown by case 2a followed by case 2b to show $\mathsf{add}(\mathsf{s}(x), y) >_{\mathsf{L}} \mathsf{add}(x, y)$.

The *Knuth–Bendix order* (KBO) [63] uses weight in addition to precedence. A *weight* is a pair (w_0, w) consisting of a positive integer w_0 and a function w from function symbols to natural numbers with $w(c) \geqslant w_0$ for all constants c. The *weight* $w(t)$ of a term t is defined inductively: $w(x) = w_0$ for variables x, and $w(t) = w(f) + \sum_i w(t_i)$ for $t = f(t_1, \ldots, t_n)$.

Definition 2.4.4. *Let (w_0, w) be a weight and $\succeq$ a partial precedence. The Knuth–Bendix order $>_{\mathsf{K}}$ is inductively defined as follows: $s >_{\mathsf{K}} t$ if $|s|_x \geqslant |t|_x$ for all variables x, and*

1. *$w(s) > w(t)$, or*
2. *$w(s) = w(t)$ and one of the following conditions holds.*
   a. *$s = f^n(t)$ for some $n > 0$ and t is a variable.*
   b. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and either*
      i. *$f \succ g$ or*
      ii. *$f = g$ and $(s_1, \ldots, s_m) >_{\mathsf{K}}^{\mathsf{lex}} (t_1, \ldots, t_n)$.*

The weight (w_0, w) is admissible *for $\succeq$, if $f \succ g$ for other function symbols g whenever f is a unary function symbol with $w(f) = 0$.*

Proposition 2.4.5 ([63]). *If $\succeq$ is well-founded and (w_0, w) is admissible for $\succeq$, then $>_{\mathsf{K}}$ is a reduction order with the subterm property.*

Example 2.4.6. *We prove that the TRS $\mathcal{R}$ of addition (Example 2.2.1) is terminating by the KBO. Consider the admissible weight $w_0 = w(0) = w(\mathsf{s}) = w(\mathsf{add}) = 1$ (which simply counts the number of symbols in a term) and a precedence $\succeq$ with $\mathsf{add} \succ \mathsf{s}$. We verify that the induced KBO $>_{\mathsf{K}}$ satisfies $\mathcal{R} \subseteq >_{\mathsf{K}}$: for the first rule, $0 + y >_{\mathsf{K}} y$ follows from case 1 as $w(0 + y) = 3 > 1 = w(y)$; for the second rule, $\mathsf{add}(\mathsf{s}(x), y) >_{\mathsf{K}} \mathsf{s}(\mathsf{add}(x, y))$ follows from case 2b as $w(\mathsf{add}(\mathsf{s}(x), y)) = 4 = w(\mathsf{s}(\mathsf{add}(x, y)))$ and $\mathsf{add} \succ \mathsf{s}$.*

The *weighted path orders* (WPO) is a class of reduction orders introduced by Yamada, Kusakari and Sakabe [117]. The definition of WPOs is based on an algebra $\mathcal{A}$ and a precedence $\succeq$.[3] A WPO compares terms s and t as a KBO does: First the terms are compared by $s >_{\mathcal{A}} t$. If only weak inequality $s \geqslant_{\mathcal{A}} t$ holds then their root symbols, say f and g, are compared by the precedence $\succeq$. If again only weak inequality $f \succeq g$ holds, arguments are compared lexicographically.

Definition 2.4.7 ([117]). *Let $\mathcal{A}$ be an ordered $\mathcal{F}$-algebra and $\succeq$ a precedence. The* weighted path order $>_{\mathsf{W}}$ *is defined on terms over $\mathcal{F}$ as follows: $s >_{\mathsf{W}} t$ if*

1. $s >_{\mathcal{A}} t$, *or*
2. $s \geqslant_{\mathcal{A}} t$, $s = f(s_1, \dots, s_m)$, *and one of the following conditions holds.*
   a. $s_i \geqslant_{\mathsf{W}} t$ *for some* $1 \leqslant i \leqslant m$.
   b. $t = g(t_1, \dots, t_n)$ *and* $s >_{\mathsf{W}} t_j$ *for all* $1 \leqslant j \leqslant n$, *and moreover*
      i. $f \succ g$, *or*
      ii. $f \succeq g$ *and* $(s_1, \dots, s_m) >_{\mathsf{W}}^{\mathsf{lex}} (t_1, \dots, t_n)$.

Here $\geqslant_{\mathsf{W}}$ denotes the reflexive closure of $>_{\mathsf{W}}$.

Theorem 2.4.8 ([117]). *Suppose that the signature is bounded. For every well-founded, normal, weakly simple and monotone algebra and well-founded precedence, the induced relation $>_{\mathsf{W}}$ is a reduction order with the subterm property. If in addition the algebra is almost total and the precedence is total, then $>_{\mathsf{W}}$ is ground total.*

We note that the normality requirement is spotted by [103]. Otherwise, the order loses monotonicity, as witnessed by the following pathological example.

Example 2.4.9. *Consider the signature $\mathcal{F} = \{\mathsf{f}^{(1)}, \mathsf{a}^{(0)}, \mathsf{b}^{(0)}\}$. As there is no function symbol of arity greater than* 1, *the relation $\trianglerighteq$ is a rewrite preorder. We define $s > t$ as $t = \mathsf{b}$ and either $s = \mathsf{a}$ or $s = \mathsf{f}(u)$ for some term u. Then $(\trianglerighteq, >)$ is a reduction pair. Now, we consider the term algebra $\bar{\mathcal{T}}$ equipped with $(\trianglerighteq, >)$ which is well-founded, weakly monotone and weakly simple, but not normal since $\mathsf{a} > \mathsf{b}$ but $\mathsf{a} \trianglerighteq \mathsf{b}$ does not hold. The WPO $>_{\mathsf{W}}$ induced by $\bar{\mathcal{T}}$ and an arbitrary precedence is not monotone: $\mathsf{a} >_{\mathsf{W}} \mathsf{b}$ by case 1 but $\mathsf{f}(\mathsf{a}) >_{\mathsf{W}} \mathsf{f}(\mathsf{b})$ does not hold because neither $\mathsf{f}(\mathsf{a}) > \mathsf{f}(\mathsf{b})$ nor $\mathsf{f}(\mathsf{a}) \trianglerighteq \mathsf{f}(\mathsf{b})$ holds.*

Weak simplicity of algebra is essential for well-foundedness of WPO. Even worse, otherwise the order is not even irreflexive.

[3] In contrast to [117, Definition 3], we do not consider the status extension to allow permutation of arguments. This is merely to simplify the presentation.

Example 2.4.10. *Let $\mathcal{A}$ be the weakly monotone and normal algebra on $\mathbb{N}$ given by $\mathsf{a}_{\mathcal{A}} = 1$ and $\mathsf{f}_{\mathcal{A}}(x) = 0$. The WPO $>_{\mathsf{W}}$ induced by $\mathcal{A}$ and any precedence satisfies $\mathsf{f}(\mathsf{a}) >_{\mathsf{W}} \mathsf{f}(\mathsf{a})$ which can be seen by applying case 2a and then case 1.*

The next is an example of termination proof by WPO.

Example 2.4.11. *Consider the following TRS $\mathcal{R}$ [117, Example 9]:*

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{f}(\mathsf{f}(x))) \qquad\qquad \mathsf{f}(\mathsf{h}(x)) \to \mathsf{h}(\mathsf{h}(\mathsf{f}(x)))$$

Let $\mathcal{A}$ be the weakly simple and monotone algebra on $\mathbb{N}$ with $\mathsf{f}_{\mathcal{A}}(x) = \mathsf{h}_{\mathcal{A}}(x) = x$ and $\mathsf{g}_{\mathcal{A}}(x) = x + 1$. Take a precedence $\succeq$ with $\mathsf{f} \succ \mathsf{g} \succ \mathsf{h}$. The relation $\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{W}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$ is verified by the following derivation:

$$\dfrac{\mathsf{f}(\mathsf{g}(x)) \geqslant_{\mathcal{A}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x))) \qquad \mathsf{f} \succ \mathsf{g} \qquad \dfrac{\mathsf{f}(\mathsf{g}(x)) >_{\mathcal{A}} \mathsf{f}(\mathsf{f}(x))}{\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{W}} \mathsf{f}(\mathsf{f}(x))}\ 1}{\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{W}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))}\ 2b(i)$$

Here 1 and 2b(i) indicate the corresponding cases in Definition 2.4.7. Similarly, one can verify $\mathsf{f}(\mathsf{h}(x)) >_{\mathsf{W}} \mathsf{h}(\mathsf{h}(\mathsf{f}(x)))$. Therefore, $\mathcal{R} \subseteq >_{\mathsf{W}}$ follows. Hence, we conclude that $\mathcal{R}$ is terminating.

In general, it is known that the WPO generalizes the LPO and the KBO in the following sense: KBOs are WPOs induced by a restricted form of linear polynomial interpretations over $\mathbb{N}$, and LPOs are WPOs induced by a restricted form of max/plus interpretations [117]. So every termination proof by an LPO or KBO (like Examples 2.4.6 and 2.5.3) can be simulated by a corresponding WPO.

We hint a limitation of termination proving by reduction orders with the subterm property. A TRS $\mathcal{R}$ on a finite signature is *simply terminating* [78, Definition 4.1] if there is a reduction order $>$ with $\mathcal{R} \subseteq >$ and the subterm property. There is a TRS that is terminating but not simply: $\{\mathsf{f}(\mathsf{f}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{f}(x)))\}$ is such an example [120]. In other words, the WPO cannot show its termination, let alone the LPO and the KBO.

2.5 Semantic Path Order

The *semantic path order* (SPO) [59] can be used for showing termination of non-simply terminating TRSs. Among many variants of it, we need a version of semantic path order based on order pair, and lexicographic comparison of arguments. Use of this version is essential for showing the correspondence between the WPO (Definition 2.4.7) and the SPO, as we see in Chapter 3.

Definition 2.5.1. *Let $(\sqsupseteq, \sqsupset)$ be a pair of relations on terms.*[4] *The* semantic path order *$>_{\mathsf{S}}$ is defined on terms as follows: $s >_{\mathsf{S}} t$ if $s = f(s_1, \ldots, s_m)$ and one of the following conditions holds.*

1. *$s_i \geqslant_{\mathsf{S}} t$ for some $1 \leqslant i \leqslant m$.*
2. *$t = g(t_1, \ldots, t_n)$ and $s >_{\mathsf{S}} t_j$ for all $1 \leqslant j \leqslant n$, and moreover*
   a. *$s \sqsupset t$, or*
   b. *$s \sqsupseteq t$ and $(s_1, \ldots, s_m) >_{\mathsf{S}}^{\mathsf{lex}} (t_1, \ldots, t_n)$.*

Here $\geqslant_{\mathsf{S}}$ denotes the reflexive closure of $>_{\mathsf{S}}$.

Remark 2.5.2. *In the literature, there is no definition of the SPO that is precisely identical to Definition 2.5.1, to our best knowledge. The standard definitions of SPOs ([59] and [22, Definition 4.1.19]) use the multiset extension instead of the lexicographic extension. The lexicographic version ([22, Definition 4.5.1]) introduced by Borralleras can be obtained by setting $\sqsupset$ to the strict part of $\sqsupseteq$ in Definition 2.5.1. In her thesis, one can find a multiset version [22, Definition 4.1.19] of ours.*

LPOs are special instances of SPOs:

Example 2.5.3. *Let $\succeq$ be a precedence. We define $s \sqsupseteq t$ as $s = t$, or $s = f(\bar{s})$, $t = g(\bar{t})$ and $f \succeq g$, and let $\sqsupset$ be the strict part of $\sqsupseteq$. The lexicographic path order $>_{\mathsf{L}}$ induced by $\succeq$ could be defined as the semantic path order induced by $(\sqsupseteq, \sqsupset)$.*

Now, we prove properties of this version of SPO. The proof techniques are also useful for the main part of the thesis.

Let $(\sqsupseteq, \sqsupset)$ be a stable order pair on $\mathcal{T}$ and let $>_{\mathsf{S}}$ be the semantic path order induced by $(\sqsupseteq, \sqsupset)$. The transitivity, reflexivity, and stability of $>_{\mathsf{S}}$ are straightforward to prove.

Lemma 2.5.4. *The SPO $>_{\mathsf{S}}$ is a stable strict order with the subterm property.* ∎

Hereafter we assume that $\sqsupset$ is well-founded and $\mathcal{F}$ is bounded. For showing that $>_{\mathsf{S}}$ is well-founded, we adopt Buchholz's method [28]. One can find a similar proof in [117, Lemma 8]. We write $\mathsf{SN}(>_{\mathsf{S}})$ for the set of all terms t such that there is no infinite descending sequence $t >_{\mathsf{S}} t_1 >_{\mathsf{S}} t_2 >_{\mathsf{S}} \cdots$ starting from t.[5] The following properties are immediate:

- The restriction of $>_{\mathsf{S}}$ to $\mathsf{SN}(>_{\mathsf{S}})$ is a well-founded order on $\mathsf{SN}(>_{\mathsf{S}})$.
- $t \in \mathsf{SN}(>_{\mathsf{S}})$ if $u \in \mathsf{SN}(>_{\mathsf{S}})$ for all terms u with $t >_{\mathsf{S}} u$.

[4] Observe that Definition 2.5.1 uses the order pair only when s and t are not variables, so behavior of the orders on variables does not matter.

[5] SN stands for strong normalization, which is another name of termination.

Buchholz's method proves well-foundedness by well-founded induction. The lexicographic product $\gg$ given by

$$(\sqsupseteq, \sqsupset) \otimes (\geqslant_{\mathsf{S}}^{\mathsf{lex}}, >_{\mathsf{S}}^{\mathsf{lex}}) \otimes (\trianglerighteq, \triangleright)$$

is a well-founded order on $\mathcal{T} \times \mathsf{SN}(>_{\mathsf{S}})^{\leqslant M} \times \mathcal{T}$. Here M stands for the maximum arity in the signature $\mathcal{F}$, and $\geqslant_{\mathsf{S}}^{\mathsf{lex}}$ for the reflexive closure of $>_{\mathsf{S}}^{\mathsf{lex}}$.

Lemma 2.5.5. *The term u belongs to $\mathsf{SN}(>_{\mathsf{S}})$ whenever $t = f(t_1, \ldots, t_n) >_{\mathsf{S}} u$ and $t_1, \ldots, t_n \in \mathsf{SN}(>_{\mathsf{S}})$.*

Proof. We show the claim by well-founded induction on $(t, (t_1, \ldots, t_n), u)$ with respect to $\gg$. Here we proceed by analyzing the derivation of $t >_{\mathsf{S}} u$. If $t >_{\mathsf{S}} u$ is derived from SPO 1 then $t_i \geqslant_{\mathsf{S}} u$ for some $i \in \{1, \ldots, n\}$. In this case $u \in \mathsf{SN}(>_{\mathsf{S}})$ trivially follows from $t_i \in \mathsf{SN}(>_{\mathsf{S}})$. If $t >_{\mathsf{S}} u$ is derived from SPO 2a or SPO 2b, then u is of the form $g(u_1, \ldots, u_m)$ and $t >_{\mathsf{S}} u_j$ for all $j \in \{1, \ldots, m\}$. From $u \triangleright u_j$ we have $(t, (t_1, \ldots, t_n), u) \gg (t, (t_1, \ldots, t_n), u_j)$. So from the induction hypothesis $u_i \in \mathsf{SN}(>_{\mathsf{S}})$ for each j. For showing our goal $u \in \mathsf{SN}(>_{\mathsf{S}})$ fix an arbitrary term v with $u >_{\mathsf{S}} v$. We further distinguish the case of SPO 2a and that of SPO 2b.

a. If $t >_{\mathsf{S}} u$ is derived from SPO 2a then $t \sqsupset u$. Thus, $(t, (t_1, \ldots, t_n), u) \gg (u, (u_1, \ldots, u_m), v)$, and the induction hypothesis yields $v \in \mathsf{SN}(>_{\mathsf{S}})$.

b. If $t >_{\mathsf{S}} u$ is derived from SPO 2b then we additionally have $t \sqsupseteq u$ and $(t_1, \ldots, t_n) >_{\mathsf{S}}^{\mathsf{lex}} (u_1, \ldots, u_m)$. Thus, $(t, (t_1, \ldots, t_n), u) \gg (u, (u_1, \ldots, u_m), v)$ holds. So from the induction hypothesis we obtain $v \in \mathsf{SN}(>_{\mathsf{S}})$.

In either case $v \in \mathsf{SN}(>_{\mathsf{S}})$. So we conclude $u \in \mathsf{SN}(>_{\mathsf{S}})$. ∎

Lemma 2.5.6. *The relation $>_{\mathsf{S}}$ is well-founded.*

Proof. We show that $t \in \mathsf{SN}(>_{\mathsf{S}})$ by induction on $|t|$. If t is a variable trivially $t \in \mathsf{SN}(>_{\mathsf{S}})$. Otherwise, Lemma 2.5.5 applies. ∎

Theorem 2.5.7. *Suppose that the signature is bounded. If $(\sqsupseteq, \sqsupset)$ is a stable and well-founded order pair on terms, the induced semantic path order is a stable well-founded order with the subterm property.*

In general, semantic path orders are not closed under contexts. For a remedy, Borralleras and her collaborators [22, 26] have proposed the use of another preorder with the *harmony* property. This results in *monotonic semantic path orders* (MSPOs).

Definition 2.5.8 ([22, Definition 4.1.20])**.** *A triple $(\geqslant, \sqsupseteq, \sqsupset)$ is a* reduction triple *if $\geqslant$ is a rewrite preorder on terms, $(\sqsupseteq, \sqsupset)$ is a stable order pair on $\mathcal{T}$ with $\sqsupset$ well-founded, and $\sqsupseteq$ is harmonious to $\geqslant$.*

We add a few words for disambiguation: Borralleras calls reduction triples by reduction *triplets* [22]. In addition, in the context of dependency pairs, there is another usage of the term in a different meaning [55].

Definition 2.5.9. *Let* $(\geqslant, \sqsupseteq, \sqsupset)$ *be a reduction triple, and let* $>_{\mathsf{S}}$ *be the semantic path order induced from* $(\sqsupseteq, \sqsupset)$*. The* monotonic semantic path order $s >_{\mathsf{MS}} t$ *is defined as* $s \geqslant t$ *and* $s >_{\mathsf{S}} t$*.*

So $>_{\mathsf{MS}}$ is defined as the intersection of $\geqslant$ and $>_{\mathsf{S}}$.

Theorem 2.5.10. *Every monotonic semantic path order is a reduction order, provided that the signature is bounded.*

Proof. The proof due to Borralleras et al. [26, Theorem 2] goes through. ∎

To talk about so-called *incrementality*, we extend the notion of extension to tuples of relations. For example, we say a pair $(\sqsupseteq_A, \sqsupset_A)$ of relations extends $(\sqsupseteq_B, \sqsupset_B)$ if $\sqsupseteq_A$ extends $\sqsupseteq_B$ and also $\sqsupset_A$ extends $\sqsupset_B$. It is not hard to see that the SPO is incremental, meaning that if $(\sqsupseteq_A, \sqsupset_A)$ extends $(\sqsupseteq_B, \sqsupset_B)$ then the SPO induced by A extends that induced by B. This is because the definition uses parameters only positively.[6] By the same token, the MSPO is also incremental. As a consequence, since the greater relations are the more profitable for termination proving (see Proposition 2.2.2), for parameters we should use as large relations as possible.

From now on, we recall constructions of reduction triples. A precedence naturally induces a reduction triple.

Proposition 2.5.11 ([26])**.** *Let* $\succeq$ *be a precedence, and define* $s \sqsupseteq_P t$ *by* $s = t$ *or* $s = f(\bar{s})$, $t = g(\bar{t})$ *and* $f \succeq g$*, and* $\sqsupset_P$ *by the strict part of* $\sqsupseteq$*. Then* $(\mathcal{T} \times \mathcal{T}, \sqsupseteq_P, \sqsupset_P)$ *is a reduction triple.*

The proposition above with Theorem 2.5.10 leads to an alternative proof of the fact that LPOs are reduction orders (Proposition 2.4.3).

Every reduction pair can be turned into a reduction triple (and therefore into a reduction order by Theorem 2.5.10) in the following way.

Proposition 2.5.12 ([26])**.** *The triple* $(\geqslant, \geqslant, >)$ *is a reduction triple if* $(\geqslant, >)$ *is a reduction pair.*

Note that $>$ of Proposition 2.5.12 is not the strict part of $\geqslant$ in general. This essentially explains why we adopt an order-pair based formulation in Definition 2.5.1; see also Remark 2.5.2. This is most evident when they are induced by a polynomial interpretation.

[6] In contrast, the versions of the SPO formulated with a single quasi-order are not incremental.

Example 2.5.13. *Consider the linear polynomial interpretation $\mathcal{A}$ over $\mathbb{N}$ with the interpretation $\mathsf{f}_{\mathcal{A}}(x) = 2x$, and the induced reduction triple $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$. On the one hand we have $\mathsf{f}(\mathsf{f}(x)) \geqslant_{\mathcal{A}} \mathsf{f}(x)$ from $4x \geqslant 2x$ for all $x \in \mathbb{N}$, but not $\mathsf{f}(x) \geqslant_{\mathcal{A}} \mathsf{f}(\mathsf{f}(x))$ as $2x \geqslant 4x$ does not hold for $x = 1$. On the other hand $\mathsf{f}(\mathsf{f}(x)) >_{\mathcal{A}} \mathsf{f}(x)$ does not hold either, because $4x > 2x$ does not hold for $x = 0$.*

Now, we can prove the termination of $\mathcal{R} = \{\mathsf{f}(\mathsf{f}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{f}(x)))\}$ [120]: Let $\mathcal{A}$ be the linear polynomial interpretation over $\mathbb{N}$ with $\mathsf{f}_{\mathcal{A}}(x) = x + 1$ and $\mathsf{g}_{\mathcal{A}}(x) = 0$, and $>_{\mathsf{MS}}$ the MSPO induced by the reduction triple $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$. The inequality $\mathsf{f}(\mathsf{f}(x)) \geqslant_{\mathcal{A}} \mathsf{f}(\mathsf{g}(\mathsf{f}(x)))$ boils down to $x + 2 > 1$ for all $x \in \mathbb{N}$, which trivially holds. In this way, we can confirm $\mathsf{f}(\mathsf{f}(x)) >_{\mathcal{A}} t$ for all subterms t of $\mathsf{f}(\mathsf{g}(\mathsf{f}(x)))$. So, $\mathsf{f}(\mathsf{f}(x)) >_{\mathsf{S}} \mathsf{f}(\mathsf{g}(\mathsf{f}(x)))$ follows from successive application of case 2a. Finally, we conclude $\mathcal{R} \subseteq >_{\mathsf{MS}}$ and thereby the termination.

It is known that Proposition 2.5.12 with weakly monotone algebras can be improved by incorporating marking of root symbols. A similar idea is used by the dependency pair method to be explained in Section 2.7. More precisely, we extend the original signature $\mathcal{F}$ with a distinguished copy $\mathcal{F}^\sharp$ which consists of a fresh n-ary function symbol $f^\sharp$ for each $f^{(n)} \in \mathcal{F}$. For an $(\mathcal{F} \cup \mathcal{F}^\sharp)$-algebra $\mathcal{A}$, we define $s \geqslant^\sharp_{\mathcal{A}} t$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ by $s = t$ or $s = f(\bar{s})$, $t = g(\bar{t})$ and $f^\sharp(\bar{s}) \geqslant_{\mathcal{A}} g^\sharp(\bar{t})$. Similarly, we define $s >^\sharp_{\mathcal{A}} t$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ by $s = f(\bar{s})$, $t = g(\bar{t})$ and $f^\sharp(\bar{s}) >_{\mathcal{A}} g^\sharp(\bar{t})$.

Proposition 2.5.14 ([26])**.** *For a weakly monotone and well-founded $(\mathcal{F} \cup \mathcal{F}^\sharp)$-algebra $\mathcal{A}$, the triple $(\geqslant_{\mathcal{A}}, \geqslant^\sharp_{\mathcal{A}}, >^\sharp_{\mathcal{A}})$ is a reduction triple on $\mathcal{T}(\mathcal{F}, \mathcal{V})$.*

In the setting of Proposition 2.5.14 neither $(\geqslant^\sharp_{\mathcal{A}}, >^\sharp_{\mathcal{A}})$ nor $(\geqslant_{\mathcal{A}}, >^\sharp_{\mathcal{A}})$ is a reduction order in general. To see this, consider the polynomial interpretation $\mathcal{A}$ defined as:

$$\mathsf{a}_{\mathcal{A}} = 0 \qquad \mathsf{b}_{\mathcal{A}} = 1 \qquad \mathsf{f}_{\mathcal{A}}(x) = x$$
$$\mathsf{a}^\sharp_{\mathcal{A}} = 1 \qquad \mathsf{b}^\sharp_{\mathcal{A}} = 0 \qquad \mathsf{f}^\sharp_{\mathcal{A}}(x) = x$$

Then $\geqslant^\sharp_{\mathcal{A}}$ is not monotone, as $\mathsf{a} \geqslant^\sharp_{\mathcal{A}} \mathsf{b}$ but $\mathsf{f}(\mathsf{a}) \geqslant^\sharp_{\mathcal{A}} \mathsf{f}(\mathsf{b})$ does not hold. Moreover, $(\geqslant_{\mathcal{A}}, >^\sharp_{\mathcal{A}})$ is not compatible, as $\mathsf{a} >^\sharp_{\mathcal{A}} \mathsf{b} \geqslant_{\mathcal{A}} \mathsf{a}$ but $\mathsf{a} >^\sharp_{\mathcal{A}} \mathsf{a}$ does not hold. So, Proposition 2.5.14 shows an advantage of working with reduction triples rather than reduction pairs.

Lexicographic combination (Definition 2.1.1) is useful for reduction triples.

Proposition 2.5.15. *If $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ are reduction triples, so is $(\geqslant_A \cap \geqslant_B, \sqsupseteq_{AB}, \sqsupset_{AB})$.* ∎

We note that [26, Proposition 4.3] is a special case when $\geqslant_A$ equals to $\geqslant_B$ and moreover $\sqsupset_A$ and $\sqsupset_B$ are the strict parts of $\sqsupseteq_A$ and $\sqsupseteq_B$, respectively.

Proposition 2.5.16. *The triple $(\mathcal{T} \times \mathcal{T}, \mathcal{T} \times \mathcal{T}, \varnothing)$ is a reduction triple.*

This trivial reduction triple plays the role of neutral element for Proposition 2.5.15: If $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ is the trivial one, then $(\geqslant_A \cap \geqslant_B, \sqsupseteq_{AB}, \sqsupset_{AB})$ is identical to $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$. This is actually convenient for our theory, as we see in the main chapters.

2.6 Semantic Path Order modulo AC

Associativity $(x+y)+z = x+(y+z)$ and commutativity $x+y = y+x$ are ubiquitous. Rewriting modulo associativity and commutativity (AC) is a rewriting formalism with AC built-in where, for example, the term $(x+y)+z$ is identified with $y+(x+z)$ if $+$ is designated to be associative and commutative.

Let $\mathcal{F}_{\mathsf{AC}} \subseteq \mathcal{F}$ be a designated set of binary symbols (meant to be associative and commutative). The set AC of associativity and commutative rules for $\mathcal{F}_{\mathsf{AC}}$ consists of

$$f(x,y) \to f(y,x) \qquad\qquad f(f(x,y),z) \to f(x,f(y,z))$$

for each $f \in \mathcal{F}_{\mathsf{AC}}$. The relation $\to_{\mathsf{AC}}^*$ is an equivalence relation which we denote by $=_{\mathsf{AC}}$. Termination modulo AC is defined as relative termination with respect to AC: A TRS $\mathcal{R}$ is *terminating modulo AC* if $\mathcal{R}/\mathsf{AC}$ is terminating. There is an analog of Proposition 2.2.2 for termination modulo AC. A relation $\rightsquigarrow$ is *AC-compatible* if it satisfies $s \rightsquigarrow v$ whenever $s =_{\mathsf{AC}} t \rightsquigarrow u =_{\mathsf{AC}} v$.

Proposition 2.6.1. *A TRS $\mathcal{R}$ is terminating modulo AC if and only if there is an AC-compatible reduction order $>$ with $\mathcal{R} \subseteq >$.*

A key ingredient of term orders for modulo AC is the (top) *flattening* $\triangledown(t)$ [86], which extract, as a multiset, the arguments of a term t in the sense of modulo AC. Formally, the flattening $\triangledown(t)$ of a term $t = f(t_1, \ldots, t_n)$ is a multiset of terms defined by $\triangledown(t) = \triangledown_f(t_1) \uplus \cdots \uplus \triangledown_f(t_n)$, where $\triangledown_f(u)$ is defined recursively as follows:

$$\triangledown_f(u) = \begin{cases} \triangledown_f(u_1) \uplus \triangledown_f(u_2) & \text{if } u = f(u_1,u_2) \text{ and } f \in \mathcal{F}_{\mathsf{AC}} \\ \{u\} & \text{otherwise} \end{cases}$$

For example, if the signature contains a binary symbol $+ \in \mathcal{F}_{\mathsf{AC}}$ and a unary symbol f, then $\triangledown((x+y)+\mathsf{f}(z+w)) = \{x, y, \mathsf{f}(z+w)\}$.

Borralleras [22] has proposed two AC versions of the monotonic semantic path order. Here we consider the simpler version called *E-AC-MSPO*.

Definition 2.6.2 ([22, Section 5.3])**.** *Let $(\geqslant, \sqsupseteq, \sqsupset)$ be a triple of relations on terms. The relation $s = f(\bar{s}) >_{\mathsf{ACS}} t$ is defined recursively as follows:*

1. *$s' \geqslant_{\mathsf{ACS}} t$ for some $s' \in \triangledown(s)$*

2. $t = g(\bar{t})$, $s \sqsupset t$, *and* $\{s\} >_{\mathsf{ACS}}^{\mathsf{mul}} \triangledown(t)$

3. $t = f(\bar{t})$, $s \sqsupseteq t$, *and* $\triangledown(s) >_{\mathsf{ACS}}^{\mathsf{mul}} \triangledown(t)$

Here $>_{\mathsf{ACS}}^{\mathsf{mul}}$ *is from the multiset extension of* $=_{\mathsf{AC}}$ *and* $>_{\mathsf{ACS}}$, *and* $\geqslant_{\mathsf{ACS}}$ *is the union of* $=_{\mathsf{AC}}$ *and* $>_{\mathsf{ACS}}$. *The E-AC-MSPO* $>_{\mathsf{ACMS}}$ *is defined as the intersection of* $\geqslant$ *and* $>_{\mathsf{ACS}}$.

For simplicity, in case 3 the head symbols of s and t are assumed to be the same. To extend the case for different head symbols, we need to adjust the definition of $\triangledown$ in a way that the equivalence induced by a precedence is taken into account, as remarked soon after [22, Definition 5.3.1]. This is beyond the scope of this thesis.

A relation $\rightsquigarrow$ on terms has the *AC-deletion property* if $f(f(t_1, t_2), t_3) \rightsquigarrow f(t_1, t_2)$ and $f(t_1, f(t_2, t_3)) \rightsquigarrow f(t_2, t_3)$ for all terms t_1, t_2, t_3 and $f \in \mathcal{F}_{\mathsf{AC}}$. The relation is *AC-monotone* if $f(s_1, s_2) \rightsquigarrow t$ implies $f(f(s_1, s_2), u) \rightsquigarrow f(t, u)$ for all terms s_1, s_2, t, u and $f \in \mathcal{F}_{\mathsf{AC}}$. A triple $(\geqslant, \sqsupseteq, \sqsupset)$ of relations on terms is an *AC reduction triple* if $(\geqslant, \sqsupseteq, \sqsupset)$ is a reduction triple with each relation AC-compatible, $\sqsupseteq$ has the AC-deletion property, and $\sqsupset$ is AC-monotone.

Theorem 2.6.3 ([22, Theorem 5.3.5])**.** *The E-AC-MSPO* $>_{\mathsf{ACMS}}$ *induced by an AC reduction triple* $(\geqslant, \sqsupseteq, \sqsupset)$ *is an AC-compatible reduction order.*

Since we use multiset in the AC setting, the boundedness condition of signature is not necessary.

The E-AC-MSPO is a generalization of Hirokawa and Middeldorp's AC multiset path order [57], which is obtained from an AC reduction triple induced by a precedence.

Proposition 2.6.4. *For a precedence* $\succeq$, *define* $s \sqsupseteq_P t$ *as* $s = t$ *or* $s = f(\bar{s})$, $t = g(\bar{t})$ *and* $f \succeq g$ *and define* $s \sqsupset_P t$ *as the strict part of* $\sqsupseteq$. *If every AC symbol* $f \in \mathcal{F}_{\mathsf{AC}}$ *is minimal with respect to* $\succ$, *then* $(\mathcal{T} \times \mathcal{T}, \sqsupseteq_P, \sqsupset_P)$ *is an AC reduction triple.*

Proof. This is a corollary of [22, Lemma 5.3.15]. ∎

We note that in the setting of Proposition 2.6.4 it is possible to have two different AC symbols f and g as long as neither $f \succ g$ nor $g \succ f$ holds. So, $f \sim g$ is admissible.

For example, let $\mathcal{F} = \{-^{(1)}, +^{(2)}\}$ and $\mathcal{F}_{\mathsf{AC}} = \{+\}$. The precedence $\succeq$ with $- \succ +$ satisfies the minimality condition. So, the E-AC-MSPO $>_{\mathsf{ACMS}}$ induced by Proposition 2.6.4 with $\succeq$ is an AC-compatible reduction order which satisfies $-(x + y) >_{\mathsf{ACMS}} (-y) + (-x)$ by case 2. This order $>_{\mathsf{ACMS}}$ is identical to the aforementioned AC multiset path order induced by $\succeq$.

We remark that the minimality requirement is essential for $(\mathcal{T} \times \mathcal{T}, \sqsupseteq_P, \sqsupset_P)$ to be an AC-reduction triple. Otherwise, $\sqsupset_P$ may not be AC-monotone, and

because of this, the resulting $>_{\mathsf{ACMS}}$ may not monotone for AC symbols, see [56, Section 3].

We end this section with a trivial AC reduction triple.

Proposition 2.6.5. *The triple* $(\mathcal{T} \times \mathcal{T}, \mathcal{T} \times \mathcal{T}, \varnothing)$ *is an AC reduction triple.* ∎

2.7 Dependency Pairs

We recall the *dependency pair framework* [3, 47, 54, 55]. Let $\mathcal{R}$ be a TRS. We write $\mathcal{D}_\mathcal{R}$ (or simply $\mathcal{D}$ when $\mathcal{R}$ can be inferred) for the set of *defined symbols* $\{f \mid f(\ell_1, \ldots, \ell_n) \to r \in \mathcal{R}\}$. Other symbols are called *constructor symbols*. Given a term t of the form $f(t_1, \ldots, t_n)$ with $f \in \mathcal{D}_\mathcal{R}$, we write $t^\sharp$ for $f^\sharp(t_1, \ldots, t_n)$. Here $f^\sharp$ is a fresh n-ary function symbol corresponding to f. The set of such terms $t^\sharp$ is denoted by $\mathcal{T}^\sharp$. The TRS $\mathsf{DP}(\mathcal{R})$ is defined as follows:

$$\mathsf{DP}(\mathcal{R}) = \{\ell^\sharp \to t^\sharp \mid \ell \to r \in \mathcal{R}, r \trianglerighteq t, \mathrm{root}(t) \in \mathcal{D}_\mathcal{R}, \text{ and } \ell \not\trianglerighteq t\}$$

We note that the condition $\ell \not\trianglerighteq t$ is Dershowitz's refinement [34]. Rules in $\mathsf{DP}(\mathcal{R})$ are called *dependency pairs*. *Dependency pair problems* are pairs $(\mathcal{P}, \mathcal{R})$ of TRSs with $\mathcal{P} \subseteq \mathcal{T}^\sharp \times \mathcal{T}^\sharp$ and $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{T}$. A dependency pair problem $(\mathcal{P}, \mathcal{R})$ is *finite* if there exists no infinite sequence of the form $s_1 \to_\mathcal{R}^* t_1 \xrightarrow{\epsilon}_\mathcal{P} s_2 \to_\mathcal{R}^* t_2 \xrightarrow{\epsilon}_\mathcal{P} \cdots$, where each s_i is terminating with respect to $\mathcal{R}$.

Theorem 2.7.1. *A TRS* $\mathcal{R}$ *is terminating if and only if* $(\mathsf{DP}(\mathcal{R}), \mathcal{R})$ *is finite.*

Whenever we apply the dependency pair method to a concrete TRS, we stick to the following convention: for a function symbol f in the original signature $\mathcal{F}$, the capitalized symbol F (instead of $f^\sharp$) denotes the corresponding copy introduced by the dependency pair method. This is reasonable, for example, when we use the dependency pair method with the semantic path order and Proposition 2.5.14, both of which perform marking to root symbols.

Example 2.7.2. *Let* $\mathcal{R}$ *be the following TRS (adapted from [121])*

$$\mathsf{p}(\mathsf{s}(x)) \xrightarrow{1} x \qquad\qquad \mathsf{f}(\mathsf{s}(x)) \xrightarrow{2} \mathsf{s}(x) \times \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$$

which is a part of a TRS implementation of factorial. The defined symbols are f *and* p*, so* $\mathsf{DP}(\mathcal{R}) = \{\mathsf{F}(\mathsf{s}(x)) \to \mathsf{F}(\mathsf{p}(\mathsf{s}(x))), \mathsf{F}(\mathsf{s}(x)) \to \mathsf{P}(\mathsf{s}(x))\}$.

Reduction pairs $(\gtrsim, >)$ are common decreasing measures for the dependency pair method. In fact, order-like properties such as transitivity are not essential in the method. We call such a weak form of reduction pairs by *pseudo-reduction pairs*. Formally, a pair $(\gtrsim, >)$ of relations on terms is called a pseudo-reduction pair if $\gtrsim$ and $>$ are stable and compatible, $\gtrsim$ is closed under contexts, and $>$ is

well-founded.[7] Note that compatibility of $(\geqslant, >)$ is still required. If in addition $>$ is closed under contexts, $(\geqslant, >)$ is called *monotone*. One can show that if $(\geqslant, >)$ is a pseudo-reduction pair, then $(\geqslant^*, >^+)$ is a reduction pair.

The next theorem is known as *reduction pair processor*.

Theorem 2.7.3. *Let $(\geqslant, >)$ be a pseudo-reduction pair. A dependency pair problem $(\mathcal{P}, \mathcal{R})$ with $\mathcal{P} \cup \mathcal{R} \subseteq \geqslant$ is finite if and only if $(\mathcal{P} \setminus >, \mathcal{R})$ is finite.*

Since $(\varnothing, \mathcal{R})$ is trivially finite, the theorem above implies a simple method for showing termination by a reduction pair.

Corollary 2.7.4. *A TRS $\mathcal{R}$ is terminating if there is a pseudo-reduction pair $(\geqslant, >)$ with $\mathcal{R} \subseteq \geqslant$ and $\mathsf{DP}(\mathcal{R}) \subseteq >$.*

If a monotone reduction pair is employed, Theorem 2.7.3 can be strengthened in a way that some rules in $\mathcal{R}$ can also be removed from the dependency pair problem $(\mathcal{P}, \mathcal{R})$. This technique is known as *rule removal processor*, or just rule removal.

Theorem 2.7.5. *Let $(\geqslant, >)$ be a monotone pseudo-reduction pair. A dependency pair problem $(\mathcal{P}, \mathcal{R})$ with $\mathcal{P} \cup \mathcal{R} \subseteq \geqslant$ is finite if and only if $(\mathcal{P}, \mathcal{R} \setminus >)$ is finite.*

Argument filtering [3, 68] is a popular transformation for building reduction pairs with the Knuth–Bendix order and the lexicographic path order. As the name suggests, this transformation filters out arguments from a given term. Formally, an *argument filter* π is a mapping that associates each n-ary function symbol f to an integer i or a list $[i_1, \dots, i_m]$ of integers with $1 \leqslant i_1 < \cdots < i_m \leqslant n$. If $\pi(f)$ is a list for every function symbols f, then the argument filtering is called *non-collapsing*. Non-collapsing argument filters are also called *partial statuses*.[8] The *argument filtering* $\hat{\pi}$ is defined on terms as follows:

$$\hat{\pi}(t) = \begin{cases} t & \text{if } t \text{ is a variable} \\ \hat{\pi}(t_i) & \text{if } t = f(t_1, \dots, t_n) \text{ and } \pi(f) = i \\ f(\hat{\pi}(t_{i_1}), \dots, \hat{\pi}(t_{i_m})) & \text{if } t = f(t_1, \dots, t_n) \text{ and } \pi(f) = [i_1, \dots, i_m] \end{cases}$$

Note that by applying $\hat{\pi}$, arities of function symbols may change, or some function symbols may be even eliminated. So more precisely, terms after argument filtering are of a different signature. We write $\mathcal{F}^\pi$ for it to distinguish from the original signature $\mathcal{F}$. Given a binary relation R on terms, we write $s \; R^\pi \; t$ if $\hat{\pi}(s) \; R \; \hat{\pi}(t)$.

[7] This definition is adapted in a formalization ([103, Section 2.2]) in the proof assistant Isabelle/HOL [83]. More precisely, their `redpair` and `redpair_order` correspond to our pseudo-reduction pair and reduction pair, respectively. There are many minor variations in the definition of reduction pair; see [115, Section 3] for a related discussion.

[8] There are versions of partial status that allow permutation and/or duplication of arguments [103, 114, 117]. Such a version could be considered with additional care of boundedness of arity.

Proposition 2.7.6. *Let $(\geqslant, >)$ be a pseudo-reduction pair and π an argument filter. Then $(\geqslant^{\pi}, >^{\pi})$ is a pseudo-reduction pair.*

Although Proposition 2.7.6 can be used with the WPO as reduction order (Definition 2.4.7), a dedicated version of the WPO as reduction pair is known. Let π be a partial status. We abuse $i \in \pi(f)$ to mean that i occurs in the list $\pi(f)$. Similarly, we may identify the list $\pi(f)$ and the corresponding set, because here we only consider an increasing list of argument positions for $\pi(f)$. If $\pi(f) = [1, \ldots, n]$ for all $f^{(n)} \in \mathcal{F}$, we call π *total*. Similarly, if $\pi(f) = [\,]$ for all $f \in \mathcal{F}$, we call π *empty*. Note that when the signature contains only constants, the unique partial status is both empty and total. If $\pi(f) = [i_1, \ldots, i_n]$ with $i_1 \leqslant \cdots \leqslant i_n$ then we write $(t_{i_1}, \ldots, t_{i_n})$ as $\pi(f)(t_1, \ldots, t_m)$. We say that an algebra $\mathcal{A}$ is *weakly (resp. strictly) π-simple* if the argument position i is weakly (resp. strictly) simple for all function symbols f and $i \in \pi(f)$.

Definition 2.7.7. *Let π be a partial status, $\mathcal{A}$ an algebra, and $\succeq$ a precedence (a quasi-order on function symbols). The* weighted path order *$(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$ is the pair of relations on terms defined simultaneously as follows: $s \geqslant_{\mathsf{W}} t$ if*

1. *$s >_{\mathcal{A}} t$, or*
2. *$s \geqslant_{\mathcal{A}} t$ and one of the following conditions holds:*
   a. *$s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{W}} t$ for some $i \in \pi(f)$.*
   b. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{W}} t_j$ for all $j \in \pi(g)$, and*
      i. *$f \succ g$ or*
      ii. *$f \succeq g$ and $\pi(f)(s_1, \ldots, s_m) \geqslant_{\mathsf{W}}^{\mathsf{lex}} \pi(g)(t_1, \ldots, t_n)$.*
   c. *$s \in \mathcal{V}$ and $s = t$*

The relation $>_{\mathsf{W}}$ is defined by cases 1, 2a and 2b where $\geqslant_{\mathsf{W}}^{\mathsf{lex}}$ is replaced by $>_{\mathsf{W}}^{\mathsf{lex}}$. Here $(\geqslant_{\mathsf{W}}^{\mathsf{lex}}, >_{\mathsf{W}}^{\mathsf{lex}})$ is the lexicographic extension of $(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$.

Theorem 2.7.8 ([117])**.** *Let π be a partial status, $\succeq$ a well-founded precedence, and $\mathcal{A}$ an algebra that is well-founded, normal, weakly π-simple and weakly monotone. Then $(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$ is a reduction pair. If in addition π is total, then $(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$ is a monotone reduction pair.*

The assumption of weak π-simplicity is essential for well-foundedness, cf. Example 2.4.10.

Example 2.7.9. *Consider the signature $\{\mathsf{a}^{(0)}, \mathsf{f}^{(1)}\}$, the partial status π with $\pi(\mathsf{f}) = [1]$ and the algebra $\mathcal{A}$ on $\mathbb{N}$ defined by $\mathsf{a}_{\mathcal{A}} = 1$ and $\mathsf{f}_{\mathcal{A}}(x) = 0$. The algebra is not weakly simple with respect to the partial status π, and the resulting WPO (with any precedence) is not well-founded, as $\mathsf{f}(\mathsf{a}) >_{\mathsf{W}} \mathsf{f}(\mathsf{a})$ by case 2a.*

Example 2.7.10 (Continued from Example 2.7.2)**.** *To show the termination of $\mathcal{R}$, consider the linear polynomial interpretation $\mathcal{A}$ over $\mathbb{N}$ with $\mathsf{s}_{\mathcal{A}}(x) = x + 1$ and $g_{\mathcal{A}}(x) = x$ for other function symbols g, and the partial status with $\pi(\mathsf{p}) = []$ and $\pi(g) = [1]$ for other function symbols g. The WPO $(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$ induced from $\mathcal{A}$, π and the precedence $\mathsf{s} \succ \mathsf{p} \succ \mathsf{f} \succ \mathsf{F} \succ \mathsf{P}$ satisfies $\mathcal{R} \subseteq \geqslant_{\mathsf{W}}$ and also $\mathsf{DP}(\mathcal{R}) \subseteq >_{\mathsf{W}}$. In particular, $\mathsf{p}(\mathsf{s}(x)) \geqslant_{\mathsf{W}} x$ follows from case 1, and $\mathsf{F}(\mathsf{s}(x)) >_{\mathsf{W}} \mathsf{F}(\mathsf{p}(\mathsf{s}(x)))$ is by case 2b(ii) followed by case 2b(i), where the comparison $\mathsf{s}(x) >_{\mathsf{W}} \mathsf{s}(x)$ is avoided thanks to $\pi(\mathsf{p}) = []$. So we conclude that $\mathcal{R}$ is terminating.*

2.8 Conditional Rewriting and Reachability

Chapter 5 is concerned with an order-based method for analyzing a form of the reachability problem in rewriting, which asks, informally speaking, if a term rewrites to another (after a suitable instantiation). This section provides preliminary knowledge for that. In fact, we consider this problem under a more generalized notion of rewriting known as conditional rewriting. Although there are many styles for formulating it (see [85, Chapter 7] for a detailed account), we follow the one used in [114, Chapter 6] because Chapter 5 is concerned with extending his techniques.

A *reachability atom* $s \twoheadrightarrow t$ is a pair (s, t) of terms. A *conditional rewrite rule* $\ell \to r \Leftarrow \phi$ consists of a pair (ℓ, r) of terms and a finite set ϕ of reachability atoms. So every TRS can be considered a conditional rewrite system by regarding each rewrite rule $\ell \to r$ as $\ell \to r \Leftarrow \varnothing$. For brevity, $\ell \to r \Leftarrow \varnothing$ is simply written as $\ell \to r$. We however note that, unlike rewrite rule defined in Section 2.2, we only require (ℓ, r) to be a pair of terms (without any conditions on variables). A *conditional term rewrite system* $\mathcal{R}$ is a set of conditional rewrite rules. The rewriting relation $\to_{\mathcal{R}}$ is defined as the least relation with the following property: $s \to_{\mathcal{R}} t$ if there are a conditional rewrite rule $\ell \to r \Leftarrow \phi$ in $\mathcal{R}$, a context C and a substitution σ such that $s = C[\ell\sigma]$, $t = C[r\sigma]$, and $u\sigma \to_{\mathcal{R}}^{*} v\sigma$ for all $u \twoheadrightarrow v \in \phi$. The *reachability problem* is the decision problem where, given a conditional TRS $\mathcal{R}$ and finite set ϕ of reachability atoms, we are asked to determine existence of a substitution σ such that $s\sigma \to_{\mathcal{R}}^{*} t\sigma$ for all $s \twoheadrightarrow t \in \phi$. If such a substitution exists, ϕ is said to be *$\mathcal{R}$-satisfiable*; otherwise it is *$\mathcal{R}$-unsatisfiable* (or infeasible [74]). So, the definition of $\to_{\mathcal{R}}$ states that a conditional rule $\ell \to r \Rightarrow \phi$ can rewrite $C[\ell\sigma]$ to $C[r\sigma]$ only if σ witnesses satisfiability of ϕ. Hereafter, for brevity, we may identify a reachability atom $s \twoheadrightarrow t$ and the singleton $\{s \twoheadrightarrow t\}$.

Example 2.8.1 (taken from [61])**.** *We consider the TRS $\mathcal{R}$*

$$\mathsf{a}(0, y) \to \mathsf{s}(y) \qquad \mathsf{a}(\mathsf{s}(x), 0) \to \mathsf{a}(x, \mathsf{s}(0)) \qquad \mathsf{a}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{a}(x, \mathsf{a}(\mathsf{s}(x), y))$$

of the Ackermann function. The reachability atom $\mathsf{a}(x, \mathsf{s}(y)) \twoheadrightarrow \mathsf{s}(\mathsf{s}(0))$ is satisfiable,

which is witnessed by the substitution $\sigma = \{x \mapsto \mathsf{0}, y \mapsto \mathsf{0}\}$. *On the other hand,* $\mathsf{a}(x, \mathsf{s}(y)) \twoheadrightarrow \mathsf{a}(z, y)$ *is unsatisfiable as we prove later.*

The next proposition (called *tupling*) allows us to transform a finite set of reachability atoms into a singleton in a satisfiability preserving way (see [114]).

Proposition 2.8.2. *Let* $\mathcal{R}$ *be a conditional TRS and* ϕ *a finite set* $\{s_1 \twoheadrightarrow t_1, \ldots, s_n \twoheadrightarrow t_n\}$ *of reachability atoms. Then* ϕ *is* $\mathcal{R}$*-satisfiable if and only if* $\mathsf{tp}(s_1, \ldots, s_n) \twoheadrightarrow \mathsf{tp}(t_1, \ldots, t_n)$ *is* $\mathcal{R}$*-satisfiable, where* tp *is a fresh n-ary function symbol.*

Co-rewrite pair provides a sufficient condition for unsatisfiability in terms of orders. Formally, a pair $(\geqslant, >)$ of relations on terms is a co-rewrite pair if $\geqslant$ is a rewrite preorder, $>$ is stable and $\geqslant \cap < = \varnothing$.

Proposition 2.8.3 ([114, Proposition 4])**.** *Let* $\mathcal{R}$ *be a conditional TRS. A reachability atom* $s \twoheadrightarrow t$ *is* $\mathcal{R}$*-unsatisfiable if there is a co-rewrite pair* $(\geqslant, >)$ *such that* $t > s$ *and, for all* $\ell \to r \Leftarrow \sigma$ *in* $\mathcal{R}$*, it holds that* $\ell \geqslant r$ *or* $v > u$ *for some* $u \twoheadrightarrow v \in \phi$.

In fact, if $\mathcal{R}$ is a TRS, the converse also holds ([114, Theorem 2]): there is a co-rewrite pair such that $t > s$ and $\mathcal{R} \subseteq \geqslant$ if $s \twoheadrightarrow t$ is $\mathcal{R}$-unsatisfiable.

A co-rewrite pair $(\geqslant, >)$ is *rewrite pair* if it is an order pair. Putting it another way, rewrite pair is reduction pair without well-foundedness. So, every reduction pair is a rewrite pair, and therefore a co-rewrite pair.

Recall that algebras are a typical ingredient for reduction pairs, see Proposition 2.3.1. As one can expect, this is also the case for rewrite pairs [114].

Proposition 2.8.4. *If* $\mathcal{A}$ *is a weakly monotone algebra, then* $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ *is a rewrite pair.*

Note that $\mathcal{A}$ need not be well-founded. So, for example, one can use an algebra whose carrier is the set $\mathbb{Z}_{\leqslant 0}$ of negative integers (including 0 and ordered as usual) for constructing a rewrite pair.

Example 2.8.5 ([61, Example 4.5])**.** *We prove that* $\mathsf{a}(x, \mathsf{s}(y)) \twoheadrightarrow \mathsf{a}(z, y)$ *is unsatisfiable with respect to* $\mathcal{R}$ *in Example 2.8.1. To this end, we use the weakly monotone algebra* $\mathcal{A}$ *on* $\mathbb{Z}_{\leqslant 0}$ *defined by* $\mathsf{a}_{\mathcal{A}}(x, y) = y - 1$, $\mathsf{s}_{\mathcal{A}}(x) = x - 1$ *and* $\mathsf{0}_{\mathcal{A}} = 0$. *The rewrite pair* $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ *satisfies* $\mathcal{R} \subseteq \geqslant_{\mathcal{A}}$ *and* $\mathsf{a}(z, y) >_{\mathcal{A}} \mathsf{a}(x, \mathsf{s}(y))$*, which boils down to the valid inequality* $y - 1 > y - 2$. *So we conclude that the atom is unsatisfiable from Proposition 2.8.3.*

Chapter 3

Generalized Weighted Path Order as Reduction Order

We introduce a generalization of the reduction order version of the weighted path order (Theorem 2.4.8), in the spirit of the semantic path order. As the semantic path order has the monotonic variant (Theorem 2.5.10), the generalized weighted path order also has a monotone version, dubbed monotonic generalized weighted path order. Interestingly, the (monotonic) generalized weighted path order and the (monotonic) semantic path order simulate each other. These results are not only of theoretical interest. We demonstrate that the generalized version of the WPO is more powerful than the original WPO.

3.1 Generalizing WPO

We generalize the WPO in the spirit of Kamin and Lévy. Their idea to obtain the SPO is to generalize precedence of the LPO (Definition 2.4.1) to order pairs on terms. Following this, we generalize the WPO so as to take two order pairs on terms: one takes the role of algebra, and the other that of precedence.

Definition 3.1.1. *Let $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ be pairs of relations on terms. The* generalized weighted path order $>_{\mathsf{G}}$ *(GWPO) is defined as follows: $s >_{\mathsf{G}} t$ if*

1. *$s \sqsupset_A t$, or*
2. *$s = f(s_1, \ldots, s_m) \sqsupseteq_A t$ and one of the following conditions holds:*
   a. *$s_i \geqslant_{\mathsf{G}} t$ for some argument s_i.*
   b. *$t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{G}} t_j$ for all arguments t_j, and moreover*
      i. *$f(s_1, \ldots, s_m) \sqsupset_B g(t_1, \ldots, t_n)$, or*
      ii. *$f(s_1, \ldots, s_m) \sqsupseteq_B g(t_1, \ldots, t_n)$ and $(s_1, \ldots, s_m) >_{\mathsf{G}}^{\mathsf{lex}} (t_1, \ldots, t_n)$.*

It is nice exercise to compare Definition 3.1.1 and Definition 2.5.1: an SPO induced by a pair $(\sqsupseteq, \sqsupset)$ is identical to the GWPO induced by $(\mathcal{T} \times \mathcal{T}, \varnothing)$ and

$(\sqsupseteq, \sqsupset)$. It is also instructive to compare Definition 3.1.1 and Definition 2.4.7: The order pair $(\sqsupseteq_A, \sqsupset_A)$ corresponds to $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ induced by an algebra $\mathcal{A}$, and $(\sqsupseteq_B, \sqsupset_B)$ corresponds to comparison by a precedence. Interestingly, the version of WPO formalized in the proof assistant Isabelle/HOL ([103, Section 3]) is based on a similar idea, but does not generalize precedence to order pair on terms.

Not so surprisingly, a sufficient condition for a GWPO to be a stable and well-founded order looks like a merger of that for an SPO (Theorem 2.5.7) and that for a WPO (Theorem 2.4.8).

Theorem 3.1.2. *Suppose that the signature is bounded. If $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are well-founded stable order pairs and $\sqsupseteq_A$ has the subterm property, the induced relation $>_{\mathsf{G}}$ is a stable and well-founded order with the subterm property.*

We prove the theorem in the next section, as a corollary of the main simulation result. The subterm property of $\sqsupseteq_A$ is essential for well-foundedness, cf. Example 2.4.10.

As the SPO has the monotonic variant MSPO, so does the GWPO. We call it the *monotonic generalized weighted path order*, or MGWPO for short.

Definition 3.1.3. *Let $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ be triples of relations on terms. The* monotonic generalized weighted path order *$>_{\mathsf{MG}}$ is the intersection $\geqslant_A \cap \geqslant_B \cap >_{\mathsf{G}}$ where $>_{\mathsf{G}}$ is induced by $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$.*

Needless to say, the MGWPO as well as the GWPO are incremental. Now, we arrive at the main theorem of this section.

Theorem 3.1.4. *Suppose that the signature is bounded. Then $>_{\mathsf{MG}}$ is a reduction order for reduction triples $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ with $\sqsupseteq_A$ having the subterm property.*

Hereafter, for brevity we may simply write the subscript A for a pair $(\sqsupseteq_A, \sqsupset_A)$ or a triple $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$, if it does not cause any confusion.

We also prove the theorem in the next section, as a corollary of the main simulation result. Instead, we demonstrate termination proving by Proposition 2.2.2 and Theorem 3.1.4. The example of TRS below is terminating but not simply terminating, so cannot be handled by WPOs. To construct reduction triples, we use Proposition 2.5.12 with well-founded and weakly monotone algebras.

Example 3.1.5. *Let $\mathcal{R}$ be the same TRS as Example 2.7.2.*

$$\mathsf{p}(\mathsf{s}(x)) \xrightarrow{1} x \qquad\qquad \mathsf{f}(\mathsf{s}(x)) \xrightarrow{2} \mathsf{s}(x) \times \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$$

Let $\mathcal{A}$ and $\mathcal{B}$ be the following max/plus algebra on $\mathbb{N}$:

$$\begin{array}{llll} \mathsf{p}_{\mathcal{A}}(x) = x & \mathsf{s}_{\mathcal{A}}(x) = x+1 & \mathsf{f}_{\mathcal{A}}(x) = x & x \times_{\mathcal{A}} y = \max\{x, y\} \\ \mathsf{p}_{\mathcal{B}}(x) = \max\{0, x-1\} & \mathsf{s}_{\mathcal{B}}(x) = x+1 & \mathsf{f}_{\mathcal{B}}(x) = x+1 & x \times_{\mathcal{B}} y = 0 \end{array}$$

The algebra $\mathcal{A}$ is well-founded, and weakly simple and monotone, and $\mathcal{B}$ is well-founded and weakly monotone. So, from Theorem 3.1.4 the MGWPO $>_{\mathsf{MG}}$ induced by the reduction triples $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\geqslant_{\mathcal{B}}, \geqslant_{\mathcal{B}}, >_{\mathcal{B}})$ is a reduction order. It remains to confirm $\mathcal{R} \subseteq >_{\mathsf{MG}}$. Checking $\mathcal{R} \subseteq \geqslant_{\mathcal{A}}$ and $\mathcal{R} \subseteq \geqslant_{\mathcal{B}}$ is easy, while $\mathcal{R} \subseteq >_{\mathsf{G}}$ is more thrilling. For rule 1, $\mathsf{p}_{\mathcal{A}}(\mathsf{s}_{\mathcal{A}}(x)) = x + 1 > x$ yields $\mathsf{p}(\mathsf{s}(x)) >_{\mathsf{G}} x$ by case 1. For rule 2, having $\mathsf{f}(\mathsf{s}(x)) \geqslant_{\mathcal{A}} \mathsf{s}(x) \times \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$ and $\mathsf{f}(\mathsf{s}(x)) >_{\mathcal{B}} \mathsf{s}(x) \times \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$ we apply case 2b(i). Then we have to verify $\mathsf{f}(\mathsf{s}(x)) >_{\mathsf{G}} \mathsf{s}(x)$ and $\mathsf{f}(\mathsf{s}(x)) >_{\mathsf{G}} \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$. The former easily follows from case 1. To see the latter, we confirm $\mathsf{f}(\mathsf{s}(x)) \geqslant_{\mathcal{A}} \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$, and $\mathsf{f}(\mathsf{s}(x)) >_{\mathcal{B}} \mathsf{f}(\mathsf{p}(\mathsf{s}(x)))$ thanks to $\mathsf{p}_{\mathcal{B}}(x) = \max\{0, x - 1\}$. The last obligation is to show $\mathsf{f}(\mathsf{s}(x)) >_{\mathsf{G}} \mathsf{p}(\mathsf{s}(x))$, which immediately follows from case 1.

The next example illustrates an MGWPO with Proposition 2.5.14.

Example 3.1.6. *Let $\mathcal{R}$ be the TRS (from [96]) consisting of the rules for group*

$$\begin{aligned} x + (-x) &\xrightarrow{1} 0 & -(-x) &\xrightarrow{6} x \\ (-x) + x &\xrightarrow{2} 0 & (x + y) + z &\xrightarrow{7} x + (y + z) \\ x + 0 &\xrightarrow{3} x & x + ((-x) + y) &\xrightarrow{8} y \\ 0 + x &\xrightarrow{4} x & (-x) + (x + y) &\xrightarrow{9} y \\ -0 &\xrightarrow{5} 0 & -(x + y) &\xrightarrow{10} (-y) + (-x) \end{aligned}$$

and the following rules for commuting endomorphisms (f and g).

$$\begin{aligned} \mathsf{f}(0) &\xrightarrow{11} 0 & -\mathsf{f}(x) &\xrightarrow{16} \mathsf{f}(-x) \\ \mathsf{g}(0) &\xrightarrow{12} 0 & -\mathsf{g}(x) &\xrightarrow{17} \mathsf{g}(-x) \\ \mathsf{f}(x) + \mathsf{f}(y) &\xrightarrow{13} \mathsf{f}(x + y) & \mathsf{f}(x) + (\mathsf{f}(y) + z) &\xrightarrow{18} \mathsf{f}(x + y) + z \\ \mathsf{g}(x) + \mathsf{g}(y) &\xrightarrow{14} \mathsf{g}(x + y) & \mathsf{g}(x) + (\mathsf{g}(y) + z) &\xrightarrow{19} \mathsf{g}(x + y) + z \\ \mathsf{f}(x) + \mathsf{g}(y) &\xrightarrow{15} \mathsf{g}(y) + \mathsf{f}(x) & \mathsf{f}(x) + (\mathsf{g}(y) + z) &\xrightarrow{20} \mathsf{g}(y) + (\mathsf{f}(x) + z) \end{aligned}$$

A difficulty of applying Proposition 2.2.2 to this example is rule 15, a terminating instance of the non-terminating rule $x + y \to y + x$. We consider the linear polynomial interpretation $\mathcal{A}$ over $\mathbb{N}$

$$0_{\mathcal{A}} = 1 \quad \mathsf{f}_{\mathcal{A}}(x) = x + 1 \quad \mathsf{g}_{\mathcal{A}}(x) = x + 1 \quad -_{\mathcal{A}}(x) = x \quad x +_{\mathcal{A}} y = x + y + 1$$

and the linear polynomial interpretation $\mathcal{B}$ over $\mathbb{N}$ for the extended signature $\mathcal{F} \cup \mathcal{F}^{\sharp}$:

$$\begin{array}{lllll} 0_{\mathcal{B}} = 0 & \mathsf{f}_{\mathcal{B}}(x) = 1 & \mathsf{g}_{\mathcal{B}}(x) = 0 & -_{\mathcal{B}}(x) = x & x +_{\mathcal{B}} y = x + y + 1 \\ 0^{\sharp}_{\mathcal{B}} = 0 & \mathsf{f}^{\sharp}_{\mathcal{B}}(x) = 1 & \mathsf{g}^{\sharp}_{\mathcal{B}}(x) = 0 & -^{\sharp}_{\mathcal{B}}(x) = x + 1 & x +^{\sharp}_{\mathcal{B}} y = x \end{array}$$

From Theorem 3.1.4 the MGWPO $>_{\mathsf{MG}}$ induced by the reduction triples $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\geqslant_{\mathcal{B}}, \geqslant^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$ is a reduction order. To conclude termination with Proposition 2.2.2, we verify $\mathcal{R} \subseteq >_{\mathsf{MG}}$. Checking $\mathcal{R} \subseteq \geqslant_{\mathcal{A}}$ and $\mathcal{R} \subseteq \geqslant_{\mathcal{B}}$ is tedious routine work. Only remark here is that $-_{\mathcal{A}}(x) = -_{\mathcal{B}}(x) = x$ is demanded by the duplicating rule 10. Then it remains to verify $\mathcal{R} \subseteq >_{\mathsf{G}}$, which is not as tedious as it seems if we can observe many rules are oriented by $>_{\mathcal{A}}$. More precisely, rules 1–4, 8, 9, 11–14, 18 and 19 are oriented by $>_{\mathcal{A}}$ and therefore by case 1. Rules 5 and 6 are oriented by case 2a. For the remaining rule, we can use case 2b(i) with

$$\begin{array}{cc} (x+y) +^{\sharp} z >_{\mathcal{B}} x +^{\sharp} (y+z) & -^{\sharp}(x+y) >_{\mathcal{B}} (-y) +^{\sharp} (-x) \\ -^{\sharp}(\mathsf{f}(x)) >_{\mathcal{B}} \mathsf{f}^{\sharp}(-x) & -^{\sharp}(\mathsf{g}(x)) >_{\mathcal{B}} \mathsf{g}^{\sharp}(-x) \\ \mathsf{f}(x) +^{\sharp} \mathsf{g}(y) >_{\mathcal{B}} \mathsf{g}(y) +^{\sharp} \mathsf{f}(x) & \mathsf{f}(x) +^{\sharp} (\mathsf{g}(y)+z) >_{\mathcal{B}} \mathsf{g}(y) +^{\sharp} (\mathsf{f}(x)+z) \end{array}$$

and then case 1 for recursive comparison.

Remark 3.1.7. *We cannot naively use Proposition 2.5.14 for the first reduction triple $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ of an MGWPO, since Theorem 3.1.4 demands the subterm property of $\sqsupseteq_A$. For example, $\mathsf{f}(\mathsf{a}) \geqslant^{\sharp}_{\mathcal{A}} \mathsf{a}$ does not hold for a weakly simple algebra $\mathcal{A}$ on $\mathbb{N}$ with $\mathsf{a}_{\mathcal{A}} = 0$, $\mathsf{a}^{\sharp}_{\mathcal{A}} = 1$, and $\mathsf{f}^{\sharp}_{\mathcal{A}}(x) = x$. So a typical choice for the first component is to use $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ with an algebra $\mathcal{A}$ on the original signature, as in the examples above. (We however have more to say about this in the next section.) Moreover, if $\mathcal{A}$ is normal, then $>_{\mathsf{G}} \subseteq \geqslant_{\mathcal{A}}$ holds and therefore $s >_{\mathsf{MG}} t$ is equivalent to $s \geqslant_B t$ and $s >_{\mathsf{G}} t$.*

3.2 Simulation between GWPOs and SPOs

We prove a simulation result between GWPOs and SPOs. As a by-product, we establish the properties of (M)GWPOs via those of (M)SPOs.

It is not hard to simulate MSPOs by MGWPOs.

Proposition 3.2.1. *Let $>_{\mathsf{MS}}$ be the MSPO induced by a reduction triple $(\geqslant, \sqsupseteq, \sqsupset)$. Then $>_{\mathsf{MS}}$ is identical to $>_{\mathsf{MG}}$ induced by the reduction triples $(\mathcal{T} \times \mathcal{T}, \mathcal{T} \times \mathcal{T}, \varnothing)$ and $(\geqslant, \sqsupseteq, \sqsupset)$.* ∎

The converse is less trivial. We show that generalized WPOs are instances of SPOs by constructing a suitable order pair from parameters of GWPO, namely $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$. To this end, we define $(\sqsupseteq_{AB}, \sqsupset_{AB})$ as the lexicographic combination of $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$. In the remaining part of the section, we consider the GWPO $>_{\mathsf{G}}$ induced by $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$, and the SPO $>_{\mathsf{S}}$ induced by the corresponding order pair $(\sqsupseteq_{AB}, \sqsupset_{AB})$.

Before presenting proofs, we illustrate how the derivation of $\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{W}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$ in Example 2.4.11 is simulated by the semantic path order.

Example 3.2.2 (continued from Example 2.4.11)**.** *From* $\mathsf{f}(\mathsf{g}(x)) \geqslant_{\mathcal{A}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$ *and* $\mathsf{f} \succ \mathsf{g}$ *the inequality* $\mathsf{f}(\mathsf{g}(x)) \sqsupset_{AB} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$ *is obtained. Moreover, we also have* $\mathsf{f}(\mathsf{g}(x)) >_{\mathcal{A}} \mathsf{f}(\mathsf{f}(x))$*. Since* $\geqslant_{\mathcal{A}}$ *has the subterm property, the subterm* $\mathsf{f}(x)$ *of* $\mathsf{f}(\mathsf{f}(x))$ *also satisfies* $\mathsf{f}(\mathsf{g}(x)) >_{\mathcal{A}} \mathsf{f}(x)$*. Thus we obtain* $\mathsf{f}(\mathsf{g}(x)) \sqsupset_{AB} \mathsf{f}(\mathsf{f}(x)), \mathsf{f}(x)$*. Therefore,* $\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))$ *is verified as follows:*

$$\dfrac{\mathsf{f}(\mathsf{g}(x)) \sqsupset_{AB} \mathsf{g}(\mathsf{f}(\mathsf{f}(x))) \qquad \mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} \mathsf{f}(\mathsf{f}(x))}{\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} \mathsf{g}(\mathsf{f}(\mathsf{f}(x)))}\,2a \qquad \dfrac{\mathsf{f}(\mathsf{g}(x)) \sqsupset_{AB} \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} \mathsf{f}(x)}{\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} \mathsf{f}(\mathsf{f}(x))}\,2a \qquad \dfrac{\mathsf{f}(\mathsf{g}(x)) \sqsupset_{AB} \mathsf{f}(x) \qquad \dfrac{\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} x}{\mathsf{g}(x) \geqslant_{\mathsf{S}} x}\,1}{\mathsf{f}(\mathsf{g}(x)) >_{\mathsf{S}} x}\,2a \qquad \dfrac{x \geqslant_{\mathsf{S}} x}{\mathsf{g}(x) \geqslant_{\mathsf{S}} x}\,1$$

Similarly, $\mathsf{f}(\mathsf{h}(x)) >_{\mathsf{S}} \mathsf{h}(\mathsf{h}(\mathsf{f}(x)))$ *can be verified. Hence, the inclusion* $\mathcal{R} \subseteq >_{\mathsf{S}}$ *holds. Observe that the use of case 1 in Example 2.4.11 is replaced by successive application of case 1 and case 2a of the SPO.*

As shown in the example, the subterm property of $\sqsupseteq_A$ is a key for filling in the gap between $>_{\mathsf{S}}$ and $>_{\mathsf{G}}$.

Lemma 3.2.3. *Suppose that* $(\sqsupseteq_A, \sqsupset_A)$ *is a stable order pair with the subterm property of* $\sqsupseteq_A$*. If* $s >_{\mathsf{G}} t$ *then* $s >_{\mathsf{S}} t$*.*

Proof. We prove the claim by induction on $|s| + |t|$. Let $s >_{\mathsf{G}} t$. We analyze the derivation of $s >_{\mathsf{G}} t$.

1. Assume that $s >_{\mathsf{G}} t$ is derived from $s \sqsupset_A t$. We distinguish three cases.
   - Suppose that s is a variable. We further analyze whether t contains s or not. If so, by the subterm property of $\sqsupseteq_A$, we have $s \sqsupset_A t \sqsupseteq_A s$, which is a contradiction; otherwise, by substituting t into s we obtain $t \sqsupset_A t$, which is again a contradiction.
   - Suppose that s is not a variable while t is a variable. If t does not occur in s then by substituting s into t we obtain $s \sqsupset_A s$, a contradiction. So t occurs in s. We obtain $s >_{\mathsf{S}} t$ by the subterm property of $>_{\mathsf{S}}$.
   - Suppose that s is not a variable, and $t = g(t_1, \ldots, t_n)$. Since $\sqsupseteq_A$ has the subterm property, for every $1 \leqslant j \leqslant n$ we have $s \sqsupset_A t \sqsupseteq_A t_j$, which leads to $s >_{\mathsf{G}} t_j$ and moreover $s >_{\mathsf{S}} t_j$ by the induction hypothesis. Moreover, $s \sqsupset t$ follows from $s \sqsupset_A t$. Hence, $s >_{\mathsf{S}} t$ follows from case 2a.
2. Assume that $s = f(s_1, \ldots, s_m)$ and $s \sqsupseteq_A t$.
   a. Suppose that $s >_{\mathsf{G}} t$ is derived from $s \sqsupseteq_A t$ and $s_i \geqslant_{\mathsf{G}} t$ for some i. Then $s >_{\mathsf{S}} t$ is obtained by case 1 using the induction hypothesis.

b. Suppose that $t = g(t_1, \ldots, t_n)$ and $s >_{\mathsf{G}} t_j$ for all $1 \leqslant j \leqslant n$. By the induction hypothesis $s >_{\mathsf{S}} t_j$ for all $1 \leqslant j \leqslant n$.

   i. If $s >_{\mathsf{G}} t$ is derived from case 2a(i), then we have $s \sqsupset_{AB} t$. So $s >_{\mathsf{S}} t$ is obtained by case 2a.

   ii. If $s >_{\mathsf{G}} t$ is derived from case 2a(ii), then we have $s \sqsupseteq_{AB} t$ and $(s_1, \ldots, s_m) >_{\mathsf{S}}^{\mathsf{lex}} (t_1, \ldots, t_n)$ by the induction hypothesis. So $s >_{\mathsf{S}} t$ is obtained by case 2b. ∎

Next we prove the converse direction of Lemma 3.2.3.

Lemma 3.2.4. *Suppose that $(\sqsupseteq_A, \sqsupset_A)$ is an order pair with the subterm property of $\sqsupseteq_A$. If $s >_{\mathsf{S}} t$ then $s >_{\mathsf{G}} t$.*

Proof. We prove the claim by induction on $|s| + |t|$. Let $s = f(s_1, \ldots, s_m)$. We analyze the derivation of $s >_{\mathsf{S}} t$.

1. Suppose that $s >_{\mathsf{S}} t$ is derived by case 1. The induction hypothesis yields $s_i \geqslant_{\mathsf{G}} t$ for some i. Moreover, by definition, $s_i \sqsupseteq_A t$ or $s_i \sqsupset_A t$ holds. In the former case, we obtain $s >_{\mathsf{G}} t$ by case 2a. In the latter case, by assumption, we have $s \sqsupseteq_A s_i \sqsupset_A t$ and therefore $s >_{\mathsf{G}} t$ by case 1.

2. Suppose that $s >_{\mathsf{S}} t$ is derived by case 2. Let $t = g(t_1, \ldots, t_n)$ and assume $s >_{\mathsf{S}} t_j$ for all $1 \leqslant j \leqslant n$. By the induction hypothesis we have $s >_{\mathsf{G}} t_j$ for all $1 \leqslant j \leqslant n$.

   a. If $s \sqsupset_{AB} t$, we further analyze how it follows: If $s \sqsupset_A t$ then we have $s >_{\mathsf{G}} t$ by case 1. Otherwise, we conclude with case 2a(i).

   b. If $s \sqsupseteq_{AB} t$, we further analyze how it follows: we further analyze how it follows: If $s \sqsupset_A t$ then we have $s >_{\mathsf{G}} t$ by case 1. Otherwise, we conclude with case 2a(ii) using the induction hypothesis. ∎

As a consequence, $>_{\mathsf{W}}$ and $>_{\mathsf{S}}$ coincide under the preconditions of Theorem 3.1.2. This proves Theorem 3.1.2 via Theorem 2.5.7. Actually, this result can be extended to the monotonic versions.

Theorem 3.2.5. *Let $>_{\mathsf{MG}}$ be the monotone generalized weighted path order induced by reduction triples $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ with the subterm property of $\sqsupseteq_A$. Then $>_{\mathsf{MG}}$ and $>_{\mathsf{MS}}$ are identical, where $>_{\mathsf{MS}}$ is induced by $(\geqslant_A \cap \geqslant_B, \sqsupseteq_{AB}, \sqsupset_{AB})$.*

Proof. Use Lemmas 3.2.3 and 3.2.4. ∎

Together with Theorem 2.5.10 and Proposition 2.5.15, this simulation result proves Theorem 3.1.4. We remark that the assumptions of Theorem 3.2.5 could be reduced. For example, well-foundedness of the reduction triples is not necessary for simulation.

It is worth examining this simulation result a little more. The MGWPO (Theorem 3.1.4) provides a way to turn two reduction triples, say $(\succcurlyeq_A, \sqsupseteq_A, \sqsupset_A)$ and $(\succcurlyeq_B, \sqsupseteq_B, \sqsupset_B)$, into a reduction order, provided that $\sqsupseteq_A$ has the subterm property. If this property is not satisfied (cf. Remark 3.1.7), as in the proof of the simulation result, we can instead use Proposition 2.5.15 to obtain the single reduction triple $(\succcurlyeq_A \cap \succcurlyeq_B, \sqsupseteq_{AB}, \sqsupset_{AB})$ then turn it into a reduction order via the MSPO (Theorem 2.5.10), which can in turn regarded as an MGWPO (Proposition 3.2.1). Bypassing the requirement of $\sqsupseteq_A$ in this way gives us more termination proving power. This is discussed in Section 3.5.

Remark 3.2.6. *We remark about the preliminary paper [87] of this chapter. There, the main result is a special case of Theorem 3.2.5 corresponding to the original WPO, namely when A is induced by a weakly simple and monotone algebra and B by a precedence (Proposition 2.5.11). Based on this result, a generalized version of the WPO is obtained by using Proposition 2.5.14 with possibly non-weakly-simple algebras for A. Somewhat confusingly, although the generalization is merely a special version of the MSPO, they call it the GWPO.*

Another way of seeing this simulation result is an optimization technique of term comparison. Unlike the SPO, the GWPO allows us to avoid recursive comparison with arguments of t if $s \sqsupset_A t$, provided that $\sqsupseteq_A$ has the subterm property. This is reminiscent of the relationship between the KBO and the WPO: the former is a special case of the latter with the strict simplicity of algebras, where recursive comparison with arguments t can be further avoided [117], cf. case 2b of Definition 2.4.4 and Definition 2.4.7. As one can expect, we also have a more KBO-like variant of GWPO that assumes the subterm property of $\sqsupset_A$ and thereby avoids unnecessary comparison.

To present the KBO-like variant, we need a few definitions. First, following [113, Definition 3.2], we extend admissibility of weight for order pairs: an order pair $(\sqsupseteq_A, \sqsupset_A)$ is *admissible* for another order pair $(\sqsupseteq_B, \sqsupset_B)$ if one of the following conditions holds for all n-ary function symbols f.

- $f(t_1, \ldots, t_n) \sqsupset_A t_i$ for all terms $t_1, \ldots, t_n$ and integers $1 \leqslant i \leqslant n$
- $n = 1$ and $f(t) \sqsupseteq_B g(\bar{u})$ for all terms $t, \bar{u}$ and function symbols g

If the latter holds for a function symbol f, we call it a *special unary symbol*. By definition, if $\sqsupset_A$ has the subterm property then A is admissible regardless of B. Moreover, we say that an order pair $(\sqsupseteq, \sqsupset)$ is *sensible* if $\mathcal{V}\mathsf{ar}(s) \supseteq \mathcal{V}\mathsf{ar}(t)$ whenever $s \sqsupseteq t$. This is a technical requirement to be explained later.

Proposition 3.2.7. *Let $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ be order pairs and $>_{\mathsf{G}}$ the GWPO induced by them. Suppose that $\sqsupseteq_A$ has the subterm property, and A is sensible and admissible for B. Then $s >_{\mathsf{G}} t$ if and only if*

1. $s \sqsupset_A t$, *or*
2. $s = f(s_1, \ldots, s_m) \sqsupseteq_A t$ *and one of the following conditions holds:*
   A. t *is a variable*[1]
   B. $t = g(t_1, \ldots, t_n)$ *and*
      I. $s \sqsupset_B t$, *or*
      II. $s \sqsupseteq_B t$ *and* $(s_1, \ldots, s_m) >_{\mathsf{G}}^{\mathsf{lex}} (t_1, \ldots, t_n)$

Proof. For the if direction, it suffices to show that each of case 1, case 2A and case 2B entails $s >_{\mathsf{G}} t$. If case 1 holds then trivially $s >_{\mathsf{G}} t$. So let $s = f(s_1, \ldots, s_m) \sqsupseteq_A t$.

A. Assume that t is a variable. Since A is sensible, t occurs in s. So $s >_{\mathsf{G}} t$ can be shown by case 1 using the subterm property of $\sqsupseteq_A$.

B. Assume that $t = g(t_1, \ldots, t_n)$.

   I. Suppose $s \sqsupset_B t$. Then g is not a special unary symbol, because otherwise we obtain a contradiction from $t \sqsupseteq_B s \sqsupset_B t$. So $s \sqsupseteq_A t \sqsupset_A t_j$ and therefore $s >_{\mathsf{G}} t_j$ for all j. Hence, $s >_{\mathsf{G}} t$ by case 2b(i).

   II. Suppose $s \sqsupseteq_B t$ and $(s_1, \ldots, s_m) >_{\mathsf{G}}^{\mathsf{lex}} (t_1, \ldots, t_n)$. If g is not a special unary symbol, then by the same argument as the previous case we can show that $s >_{\mathsf{G}} t$ by case 2b(ii). So assume that g is a special unary symbol. From $(s_1, \ldots, s_m) >_{\mathsf{G}}^{\mathsf{lex}} (t_1)$ we obtain $s_1 \geqslant_{\mathsf{G}} t_1$. Since $s >_{\mathsf{G}} s_1$ by case 2a, we now have $s >_{\mathsf{G}} t_1$ by transitivity. Hence, $s >_{\mathsf{G}} t$ by case 2b(ii).

For the only-if direction, it suffices to show that each case of $s >_{\mathsf{G}} t$ entails one of case 1, case 2A, and case 2B. Trivially, case 1 entails case 1, case 2b(i) entails case 2B(I), and case 2b(ii) entails case 2B(II). So assume that case 2a holds, namely $s = f(s_1, \ldots, s_m) \sqsupseteq_A t$ and $s_i \geqslant_{\mathsf{G}} t$ for some i. If t is a variable, then case 2A applies. So let $t = g(t_1, \ldots, t_n)$. If f is not a special unary symbol, then $s \sqsupset_A s_i \sqsupseteq_A t$ and therefore case 1 holds. Otherwise $m = i = 1$. Since $t >_{\mathsf{G}} t_1$ follows from case 2a, we obtain $s_1 >_{\mathsf{G}} t_1$ and thereby $(s_1) >_{\mathsf{G}}^{\mathsf{lex}} (t_1, \ldots, t_n)$. Hence, case 2B(II) holds. ∎

Proposition 3.2.7 allows us to improve the complexity of deciding $s >_{\mathsf{G}} t$ in the following sense: Consider deciding $s >_{\mathsf{G}} t$ for fixed order pairs A and B. If we measure the number of comparisons by A and B, the best algorithm known so far (simply following that for the LPO [73]) requires a number of comparisons quadratic in $|s| + |t|$. If fortunately A is sensible and admissible for B, Proposition 3.2.7 suggests that the number of comparisons can be reduced to

[1] This case sits oddly with case 2a of Definition 2.4.4 but goes back to [113, Definition 3.1].

linear in $|s| + |t|$. This is an analog of the fact that KBO comparison $s >_{\mathsf{K}} t$ admits an implementation of linear time complexity (with respect to $|s| + |t|$) [72].

The requirement for A to be sensible cannot be dropped because of the following pathological example: Consider the signature $\mathcal{F} = \{\mathsf{f}^{(1)}\}$. Define $s \sqsupseteq_A t$ as $s = t$ or $s = \mathsf{f}(s')$ for some term s', and $\sqsupset_A$ as $\varnothing$. Let B be the order pair induced by the precedence on $\mathcal{F}$. Then A is an order pair admissible for B, and $\sqsupseteq_A$ has the subterm property. However, the GWPO $>_{\mathsf{G}}$ induced by A and B does not satisfy $\mathsf{f}(x) >_{\mathsf{G}} y$ while case 2A follows from $\mathsf{f}(x) \sqsupseteq_A y$. (In fact, because A is a reduction pair, this example can be lifted to a counter-example to [113, Theorem 3.3] by considering the term algebra equipped with A.) Fortunately, the requirement is mild, because a typical choice for A is to use a usual interpretation (like polynomial interpretation).

Finally, we discuss a related result due to Geser that a generalized version of KBOs can be simulated by SPOs [42]. It is worth comparing his result with ours.

Definition 3.2.8 ([32, 42]). *Let $(\sqsupseteq, \sqsupset)$ be a pair of relations on terms. We define $s >_{\mathsf{D}} t$ as follows:*

1. *$s \sqsupset t$, or*

2. *$s \sqsupseteq t$, $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$ and $(s_1, \ldots, s_m) >_{\mathsf{D}}^{\mathsf{lex}} (t_1, \ldots, t_n)$.*

Here $>_{\mathsf{D}}^{\mathsf{lex}}$ is the lexicographic extension of $>_{\mathsf{D}}$.

Theorem 3.2.9 ([42]). *Let $(\sqsupseteq, \sqsupset)$ be a stable order pair with the subterm property $\rhd \subseteq \sqsupset$. Then $>_{\mathsf{D}}$ and $>_{\mathsf{S}}$ are identical, where the relations are both induced by $(\sqsupseteq, \sqsupset)$.*

In fact, our proofs for Theorem 3.2.5 and Proposition 3.2.7 owe to Geser's. We however note that Theorem 3.2.5 only uses a weaker property than $\rhd \subseteq \sqsupset$, namely $\rhd \subseteq \sqsupseteq_A$.

3.3 Ground Totality

Ground totality of a reduction order is of interest since refutational completeness of theorem proving procedures like ordered completion [10] rely on it. Unfortunately, in contrast to WPOs (Theorem 2.4.8), MGWPOs and MSPOs are not always ground total.

Example 3.3.1 (Adapted from [120]). *Let $\mathcal{F} = \{\mathsf{a}^{(0)}, \mathsf{b}^{(0)}, \mathsf{f}^{(1)}, \mathsf{g}^{(1)}\}$, and let $\mathcal{A}$ be the $\mathcal{F}$-algebra on $\mathbb{N}$*

$$\mathsf{a}_{\mathcal{A}} = 1 \qquad \mathsf{b}_{\mathcal{A}} = 1 \qquad \mathsf{f}_{\mathcal{A}}(x) = x + 1 \qquad \mathsf{g}_{\mathcal{A}}(x) = x + 1$$

and $\mathcal{B}$ the following $(\mathcal{F} \cup \mathcal{F}^\sharp)$-algebra on $\mathbb{N}$.

$$\mathsf{a}_\mathcal{B} = 1 \qquad \mathsf{b}_\mathcal{B} = 0 \qquad \mathsf{f}_\mathcal{B}(x) = x \qquad \mathsf{g}_\mathcal{B}(x) = 0$$
$$\mathsf{a}^\sharp_\mathcal{B} = 0 \qquad \mathsf{b}^\sharp_\mathcal{B} = 1 \qquad \mathsf{f}^\sharp_\mathcal{B}(x) = x \qquad \mathsf{g}^\sharp_\mathcal{B}(x) = 0$$

The MGWPO $>_\mathsf{MG}$ induced by $(\geqslant_\mathcal{A}, \geqslant_\mathcal{A}, >_\mathcal{A})$ and $(\geqslant_\mathcal{B}, \geqslant^\sharp_\mathcal{B}, >^\sharp_\mathcal{B})$ satisfies $\mathsf{f}(\mathsf{a}) >_\mathsf{MG} \mathsf{f}(\mathsf{b})$ and $\mathsf{g}(\mathsf{b}) >_\mathsf{MG} \mathsf{g}(\mathsf{a})$. If $>_\mathsf{MG}$ were ground total, then we would have either $\mathsf{a} >_\mathsf{MG} \mathsf{b}$ or $\mathsf{b} >_\mathsf{MG} \mathsf{a}$, from which we obtain a contradiction using the monotonicity. Even worse, by the same argument we can show that there is no ground-total reduction order that extends $>_\mathsf{MG}$.

An observation here is that even though a and b are incomparable with respect to $>_\mathsf{MG}$, we actually have $\mathsf{b} >_\mathsf{G} \mathsf{a}$. Indeed, we can show that $>_\mathsf{G}$ is ground total under a few reasonable assumptions.

Definition 3.3.2. *Let $(\sqsupseteq, \sqsupset)$ be a pair of relations and let $\equiv$ denote the intersection of $\sqsupseteq$ and $\sqsubseteq$. We say that $(\sqsupseteq, \sqsupset)$ is* almost ground-total *if it holds that $s \sqsupset t$, $t \sqsupset s$ or $s \equiv t$ for all ground terms s and t. We say that $(\sqsupseteq, \sqsupset)$* discriminates head symbols *if $f = g$ whenever $f(\bar{t}) \equiv g(\bar{t})$.*

Lemma 3.3.3. *Let $(\sqsupseteq_A, \sqsupset_A)$ be an almost ground-total order pair, and $(\sqsupseteq_B, \sqsupset_B)$ an almost ground-total order pair that discriminates head symbols. Then the induced GWPO $>_\mathsf{G}$ is ground-total.*

Proof. Let $s = f(s_1, \ldots, s_m)$ and $t = g(t_1, \ldots, t_n)$ be different ground terms. We prove $s >_\mathsf{G} t$ or $t >_\mathsf{G} s$ by induction on $|s| + |t|$. If $s \sqsupset_A t$ or $t \sqsupset_A s$ then we are done by case 1. So we assume $s \sqsupseteq_A t$ and $t \sqsupseteq_A s$ as $(\sqsupseteq_A, \sqsupset_A)$ is almost ground-total. If $s_i \geqslant_\mathsf{G} t$ or $t_i \geqslant_\mathsf{G} s$ for some i, then we are done by case 2a. Otherwise, by the induction hypothesis, $s >_\mathsf{G} t_j$ for all $1 \leqslant j \leqslant n$ and $t >_\mathsf{G} s_i$ for all $1 \leqslant i \leqslant m$. Then, we again use the assumption that $(\sqsupseteq_B, \sqsupset_B)$ is almost ground-total. If $s \sqsupset_B t$ or $t \sqsupset_B s$ then we are done by case 2b(i). Otherwise, $s \sqsupseteq_B t$ and $t \sqsupseteq_B s$. Since $(\sqsupseteq_B, \sqsupset_B)$ discriminates head symbols, we have $(s_1, \ldots, s_m) \neq (t_1, \ldots, t_n)$. By the induction hypothesis together with the fact that lexicographic extension preserves totality, $(s_1, \ldots, s_m) >^\mathsf{lex}_\mathsf{G} (t_1, \ldots, t_n)$ or $(t_1, \ldots, t_n) >^\mathsf{lex}_\mathsf{G} (s_1, \ldots, s_m)$, so case 2b(ii) applies.[2] ∎

Note that $(\geqslant^\sharp_\mathcal{B}, >^\sharp_\mathcal{B})$ of Example 3.3.1 does not discriminate head symbols, since both $\mathsf{f}(\mathsf{b}) \geqslant^\sharp_\mathcal{B} \mathsf{g}(\mathsf{b})$ and $\mathsf{g}(\mathsf{b}) \geqslant^\sharp_\mathcal{B} \mathsf{f}(\mathsf{b})$ hold. In such a case, we can simply combine it lexicographically with an order pair induced by a total precedence. This leads to the following result.

[2] As the proof suggests, the lemma does not hold if we adopt multiset extension in the definition of the GWPO.

Theorem 3.3.4. *Let $\succeq$ a well-founded precedence, and let $\mathcal{A}$ be an $(\mathcal{F} \cup \mathcal{F}^\sharp)$-algebra that is weakly monotone, normal, well-founded, and weakly simple for $\mathcal{F}$. Define $(\sqsupseteq, \sqsupset)$ as the lexicographic combination of $(\geqslant^\sharp_\mathcal{A}, >^\sharp_\mathcal{A})$ and the order pair induced by $\succeq$ (as in Definition 2.4.1). Then $>_\mathsf{G}$ induced by $(\geqslant_\mathcal{A}, >_\mathcal{A})$ and $(\sqsupseteq, \sqsupset)$ is a reduction order with the subterm property. If in addition $\mathcal{A}$ is almost total and $\succeq$ is total, then $>_\mathsf{G}$ is ground total.*

Proof. To show that $>_\mathsf{G}$ is a reduction order, the key is that $(\geqslant_\mathcal{A}, \geqslant_\mathcal{A}, >_\mathcal{A})$ and $(\geqslant_\mathcal{A}, \sqsupseteq, \sqsupset)$ are reduction triples with the subterm property of $\geqslant_\mathcal{A}$. Using the normality of $\mathcal{A}$, we can show that $>_\mathsf{G}$ is identical to the MGWPO induced by those reduction triples, and therefore a reduction order. Finally, Lemma 3.3.3 shows that $>_\mathsf{G}$ is ground total. ∎

The following example shows that Theorem 3.3.4 is a powerful yet ground-total extension of the original WPO.

Example 3.3.5. *Consider the system of the combinators* S *and* K *(see [11]).*

$$((\mathsf{S} \cdot x) \cdot y) \cdot z \to (x \cdot z) \cdot (y \cdot z) \qquad (\mathsf{K} \cdot x) \cdot y \to x$$

While the system is well-known to be non-terminating, the system $\mathcal{R}$ obtained by flipping the S *rule is actually terminating [16].*

$$(x \cdot z) \cdot (y \cdot z) \to ((\mathsf{S} \cdot x) \cdot y) \cdot z \qquad (\mathsf{K} \cdot x) \cdot y \to x$$

To prove the termination by Theorem 3.3.4, let $\mathcal{A}$ be the linear polynomial interpretation over $\mathbb{N}$ defined as follows:

$$x \cdot_\mathcal{A} y = x + y + 1 \qquad \mathsf{S}_\mathcal{A} = 0 \qquad \mathsf{K}_\mathcal{A} = 0$$
$$x \cdot^\sharp_\mathcal{A} y = y \qquad \mathsf{S}^\sharp_\mathcal{A} = 0 \qquad \mathsf{K}^\sharp_\mathcal{A} = 0$$

The generalized weighted path order $>_\mathsf{G}$ constructed by Theorem 3.3.4 with $\mathcal{A}$ and an arbitrary precedence satisfies $\mathcal{R} \subseteq >_\mathsf{G}$, so $\mathcal{R}$ is terminating. In particular, the comparison $(x \cdot z) \cdot (y \cdot z) >_\mathsf{G} ((\mathsf{S} \cdot x) \cdot y) \cdot z$ proceeds as follows: $(x \cdot z) \cdot (y \cdot z) \geqslant_\mathcal{A} ((\mathsf{S} \cdot x) \cdot y) \cdot z$, as $x + y + 2z + 3 \geqslant x + y + z + 3$ for all $x, y, z \in \mathbb{N}$. So we test if $(x \cdot z) \cdot^\sharp (y \cdot z) \geqslant_\mathcal{A} ((\mathsf{S} \cdot x) \cdot y) \cdot^\sharp z$, which boils down to $y + z + 1 > z$ for all $y, z \in \mathbb{N}$. The remaining obligations are $(x \cdot z) \cdot (y \cdot z) >_\mathsf{G} (\mathsf{S} \cdot x) \cdot y$ and $(x \cdot z) \cdot (y \cdot z) >_\mathsf{G} z$, which immediately follow by $>_\mathcal{A}$. We note that the termination of $\mathcal{R}$ above cannot be done by the original weighted path order if algebras are restricted to be linear polynomial interpretations over $\mathbb{N}$.

3.4 Extension to Associativity and Commutativity

The simulation result between the generalized weighted path order and the semantic path order carries over to rewriting modulo associativity and commutativity (AC), at least in a simple setting. To see this, we introduce an AC-compatible version of the generalized weighted path order which generalizes the E-AC-MSPO (Theorem 2.6.3), and then we prove a simulation result between them.

Definition 3.4.1. *Let $(\succcurlyeq_A, \sqsupseteq_A, \sqsupset_A)$ and $(\succcurlyeq_B, \sqsupseteq_B, \sqsupset_B)$ be triples of relations on terms. We define the* AC generalized weighted path order *(AC-GWPO) $>_{\mathsf{ACG}}$ by recursion: $s >_{\mathsf{ACG}} t$ if*

1. *$s \sqsupset_A t$ or*
2. *$s = f(\bar{s}) \sqsupseteq_A t$, and one of the following conditions holds.*
   a. *$s' \succcurlyeq_{\mathsf{ACG}} t$ for some $s' \in \triangledown(s)$*
   b. *$t = g(\bar{t})$, $s \sqsupset_B t$ and $\{s\} >^{\mathsf{mul}}_{\mathsf{ACG}} \triangledown(t)$*
   c. *$t = f(\bar{t})$, $s \sqsupseteq_B t$ and $\triangledown(s) >^{\mathsf{mul}}_{\mathsf{ACG}} \triangledown(t)$*

Here $>^{\mathsf{mul}}_{\mathsf{ACG}}$ is from the multiset extension of $=_{\mathsf{AC}}$ and $>_{\mathsf{ACG}}$, and $\succcurlyeq_{\mathsf{ACG}}$ is the union of $=_{\mathsf{AC}}$ and $>_{\mathsf{ACG}}$. The AC monotonic generalized weighted path order *(AC-MGWPO) $>_{\mathsf{ACMG}}$ is the intersection $\succcurlyeq_A \cap \succcurlyeq_B \cap >_{\mathsf{ACG}}$.*

Theorem 3.4.2. *For AC reduction triples $(\succcurlyeq_A, \sqsupseteq_A, \sqsupset_A)$ and $(\succcurlyeq_B, \sqsupseteq_B, \sqsupset_B)$ with AC monotonicity and the subterm property of $\sqsupseteq_A$, the relation $>_{\mathsf{ACMG}}$ is an AC-compatible reduction order.*

Later, we prove the theorem above as a corollary of a simulation result in the AC setting. A typical ingredient for $(\succcurlyeq_A, \sqsupseteq_A, \sqsupset_A)$ is algebras. An algebra $\mathcal{A}$ is *AC-compatible* if $\succcurlyeq_{\mathcal{A}}$ is AC-compatible, or equivalently, $\mathsf{AC} \subseteq \succcurlyeq_{\mathcal{A}}$; it is *strictly AC-monotone* if $\mathcal{A}$ is strictly monotone for $\mathcal{F}_{\mathsf{AC}}$.

Proposition 3.4.3. *Let $\mathcal{A}$ be a well-founded, AC-compatible, strictly AC-monotone, weakly simple and monotone algebra. Then $(\succcurlyeq_{\mathcal{A}}, \succcurlyeq_{\mathcal{A}}, >_{\mathcal{A}})$ is an AC reduction triple with AC monotonicity and the subterm property of $\succcurlyeq_A$.* ∎

Below is an example of termination proof with Theorem 3.4.2.

Example 3.4.4. *Consider the following TRS $\mathcal{R}$.*

$$\begin{aligned} 0 + x &\to x & (-x) + x &\to 0 \\ 1 \times x &\to x & (x + y) \times z &\to (x \times z) + (y \times z) \\ -(x + y) &\to (-y) + (-x) \end{aligned}$$

We prove its termination modulo AC for $\mathcal{F}_{\mathsf{AC}} = \{+, \times\}$. Let $\mathcal{A}$ be the algebra over integers $\geqslant 2$

$$0_{\mathcal{A}} = 2 \qquad 1_{\mathcal{A}} = 2 \qquad x +_{\mathcal{A}} y = x + y + 1 \qquad x \times_{\mathcal{A}} y = xy \qquad -_{\mathcal{A}}(x) = x$$

and a precedence $\succeq$ with $- \succ +$ and $+$ and $\times$ minimal. Then, we consider the AC reduction triple induced by $\mathcal{A}$ as in Proposition 3.4.3, and the one induced by $\succeq$ as in Proposition 2.6.4. By Theorem 3.4.2, the AC-MGWPO $>_{\mathsf{ACMG}}$ induced by those are an AC-compatible reduction order. So, it remains to confirm $\mathcal{R} \subseteq >_{\mathsf{ACMG}}$. For the rule $-(x+y) \to (-y)+(-x)$, we verify that both $-(x+y)$ and $(-y)+(-x)$ evaluate to $x+y+1$ in $\mathcal{A}$ and then use case 2b with $- \succ +$. The other rules can be handled by case 1.

We remark that the strict AC-monotonicity is essential for Proposition 3.4.3 and monotonicity of the thus-induced AC-MGWPO. So it is not allowed to interpret AC-symbols as max.

Example 3.4.5. *Let $\mathcal{F} = \{\mathsf{f}^{(1)}, \mathsf{g}^{(2)}, +^{(2)}\}$ and $\mathcal{F}_{\mathsf{AC}} = \{+\}$. Consider $>_{\mathsf{ACMG}}$ induced by the following algebra $\mathcal{A}$ on natural numbers*

$$\mathsf{f}_{\mathcal{A}}(x) = x + 1 \qquad \mathsf{g}_{\mathcal{A}}(x, y) = \max\{x, y\} \qquad x +_{\mathcal{A}} y = \max\{x, y\}$$

and the precedence $\mathsf{f} \succ \mathsf{g} \succ +$. Then we have $s = \mathsf{f}(x) + \mathsf{f}(y) >_{\mathsf{ACMG}} \mathsf{g}(x, y) = t$ as $\mathsf{f}(x) + \mathsf{f}(y) >_{\mathcal{A}} \mathsf{g}(x, y)$, but $s + z >_{\mathsf{ACW}} t + z$ does not hold, because the most promising case 2c demands $\triangledown(s+z) >^{\mathsf{mul}}_{\mathsf{ACG}} \triangledown(t+z)$ which is impossible.

$$\triangledown(s+z) = \{\mathsf{f}(x), \mathsf{f}(y), z\} \qquad \triangledown(t+z) = \{\mathsf{g}(x, y), z\}$$

Similarly, we cannot naively incorporate the marking technique of Proposition 2.5.14 into Proposition 3.4.3, since the resulting triple $(\geqslant_{\mathcal{A}}, \geqslant^{\sharp}_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}})$ may not satisfy AC-monotonicity of $>^{\sharp}_{\mathcal{A}}$ as well as the AC-deletion property of $\geqslant^{\sharp}_{\mathcal{A}}$.

Now, we prove Theorem 3.4.2 by appealing to Theorem 2.6.3 with the simulation technique. To this end, we need an AC version of Proposition 2.5.15.

Proposition 3.4.6. *Let $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ be AC reduction triples with AC monotonicity of $\sqsupseteq_A$. Then $(\geqslant_A \cap \geqslant_B, \sqsupseteq_{AB}, \sqsupset_{AB})$ is an AC reduction triple.*

Proof. We only prove the AC monotonicity of $\sqsupset_{AB}$, since others are straightforward to prove. Let $f(s,t) \sqsupset_{AB} u$ for an AC symbol f, and let v be a term. If $f(s,t) \sqsupset_A u$, then $f(f(s,t),v) \sqsupset_{AB} f(u,v)$ by the AC monotonicity of $\sqsupset_A$. Otherwise, $f(s,t) \sqsupseteq_A u$ and $f(s,t) \sqsupset_B u$. In this case, we use the AC monotonicity of both $\sqsupseteq_A$ and $\sqsupset_B$. ∎

From the proof, it seems that AC monotonicity of $\sqsupseteq_A$ is essential, while the author does not know a counter-example.

Proof of Theorem 3.4.2. Let $>_{\mathsf{ACMS}}$ be the E-AC-MSPO induced by the AC reduction triple $(\gtrsim_A \cap \gtrsim_B, \sqsupseteq_{AB}, \sqsupset_{AB})$. We can prove that $>_{\mathsf{ACS}}$ is identical to $>_{\mathsf{ACG}}$ by the same induction argument as in Section 3.2. So Theorem 2.6.3 shows the claim. ∎

Conversely, an E-AC-MSPO induced by $(\gtrsim, \sqsupseteq, \sqsupset)$ can be simulated by the AC-MGWPO induced by the trivial AC-reduction triple (Proposition 2.6.5) and $(\gtrsim, \sqsupseteq, \sqsupset)$.

3.5 Evaluation with Benchmark Problems

In order to evaluate our methods in termination analysis, we implemented a prototype termination tool based on Proposition 2.2.2 with the WPO, MGWPO and MSPO. We first describe the design of the experiment, and then discuss the results.

Construction of orders. The tool implements the following five kinds of constructions of reduction orders.

- $\mathtt{W}$. WPO (Theorem 2.4.8) induced by an $\mathcal{F}$-algebra $\mathcal{A}$ and a precedence
- $\mathtt{G}_1$. MGWPO (Theorem 3.1.4) induced by $(\gtrsim_{\mathcal{A}}, \gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\gtrsim_{\mathcal{B}}, \gtrsim^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$ for a weakly simple and monotone $\mathcal{F}$-algebra $\mathcal{A}$ and an $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebra $\mathcal{B}$
- $\mathtt{G}_2$. ground-total GWPO (Theorem 3.3.4) induced from an $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebra $\mathcal{A}$ weakly simple for $\mathcal{F}$ and a precedence
- $\mathtt{S}_1$. MSPO (Theorem 2.5.10) induced by the lexicographic combination of the reduction triples $(\gtrsim_{\mathcal{A}}, \gtrsim^{\sharp}_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}})$ and $(\gtrsim_{\mathcal{B}}, \gtrsim^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$ obtained from $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebras $\mathcal{A}$ and $\mathcal{B}$
- $\mathtt{S}_2$. MSPO (Theorem 2.5.10) induced by the lexicographic combination of the reduction triples obtained from an $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebra $\mathcal{A}$ and a precedence

Here $\mathcal{F}$ is the original signature of a given TRS, and $\mathcal{A}$ and $\mathcal{B}$ are linear polynomial interpretations or max/plus interpretations on natural numbers. Note that $\mathtt{S}_2$ corresponds to the GWPO in the sense of the preliminary paper, see Remark 3.2.6. In theory, for the same combination of classes of $\mathcal{A}$ and $\mathcal{B}$, the inclusions $\mathtt{W} \subsetneq X \subsetneq \mathtt{S}_1$ hold for all $X \in \{\mathtt{G}_1, \mathtt{S}_2\}$, where $\mathtt{W}$ represents the set of TRSs proven terminating by construction $\mathtt{W}$, etc. In contrast, each two of $\{\mathtt{G}_1, \mathtt{G}_2, \mathtt{S}_2\}$ are incomparable. If we use different types of algebras, even the same construction becomes incomparable, let alone different constructions. Finally, we recall that $\mathtt{W}$ and $\mathtt{G}_2$ cannot show termination beyond simple termination, while the others can.

Implementation. Our implementation uses restricted forms of interpretations similar to those used in [117, Section 7]. For a linear polynomial interpretation $f_{\mathcal{A}}(x_1, \ldots, x_n) = c_0 + c_1 x_1 + \cdots + c_n x_n$, the coefficients must satisfy $c_1, \ldots c_n \in \{0, 1\}$. This corresponds to $\mathcal{MS}$um of [117, Definition 8] whose weight status is restricted to polynomial, or strongly linear polynomial [90] where zero is allowed for coefficients. Similarly, for a max/plus interpretation $f_{\mathcal{A}}(x_1, \ldots, x_n) = \max\{c_0, c_1 + c'_1 x_1, \cdots, c_n + c'_n x_n\}$, the coefficients must satisfy $c'_1, \ldots c'_n \in \{0, 1\}$. This corresponds to $\mathcal{MS}$um of [117, Definition 8] whose weight status is restricted to max but has a couple of differences: an argument x_i may not be regarded if $c'_i = 0$, and also c_i can be negative. Following the automation techniques of KBO [119] and WPO [117], given a (finite) TRS $\mathcal{R}$, the tool finds suitable parameters (algebras and precedences) such that the induced reduction order $>$ orients all rules in $\mathcal{R}$ (i.e., $\mathcal{R} \subseteq >$) if they exist. This is done by encoding the constraints into linear arithmetic expressions and then calling the SMT solver Z3 [31] which has a decision procedure for it. See also the remarks after Propositions 2.3.2 and 2.3.4. The termination tool was run on a machine equipped with an Intel Core i7-11850H CPU (8 cores, 16 threads, 2.50–4.80 GHz) and 16 GB RAM.

Problem set. The problems for the experiment consist of 1528 finite TRSs from version 11.5 of the Termination Problem Database (TPDB) [98]. For reference, among those, the 2025 version of the termination tool AProVE (resp. NaTT) proves termination of 1028 (resp. 827) TRSs, and non-termination of 274 (resp. 169) within one minute time-limit.

How to read the result table. Let us explain how to read the result table (Table 3.1) with an example: method $\mathtt{G}_1$ whose $\mathcal{A}$ is a max/plus interpretation and $\mathcal{B}$ is a linear polynomial interpretation proved termination of 411 TRSs in the problem set, timed out for 6 problems and terminated with exceptions for 5 problems, and the total runtime was 25 minutes (after ceiling). All the exceptions in the experiment happened on large TRSs with thousands rules, so seemingly the tool terminated due to memory exhaustion. Note that methods W, $\mathtt{G}_2$ and $\mathtt{S}_2$ use a single algebra $\mathcal{A}$. The results are in the diagonal cells.

Improvement over the original WPO. Overall, use of our methods instead of the original WPO (W) significantly improves capability in termination proving. The small numbers for W and $\mathtt{G}_2$ can be explained by the fact that the problem set contains many non-simply terminating TRSs. When $\mathcal{A}$ is a linear polynomial interpretation, the improvements of $\mathtt{G}_1$ and $\mathtt{G}_2$ over W are rather mild. This is because $\ell \geqslant_{\mathcal{A}} r$ never hold for duplication rules $\ell \to r$ such as $(x + y) \times z \to (x \times z) + (y \times z)$ when $\mathcal{A}$ is required to be weakly simple. As a consequence, the

Table 3.1: Experiments on 1528 TRSs from TPDB 11.5.

		$\mathcal{B}$ is linear		$\mathcal{B}$ is max/plus
$\mathcal{A}$ is linear	$\mathtt{W}$:	*122* (10 min, 5 TO, 2 E)	$\mathtt{W}$:	-
	$\mathtt{G}_1$:	*185* (11 min, 5 TO, 2 E)	$\mathtt{G}_1$:	*201* (14 min, 2 TO, 6 E)
	$\mathtt{G}_2$:	*138* (13 min, 7 TO)	$\mathtt{G}_2$:	-
	$\mathtt{S}_1$:	*398* (20 min, 7 TO, 1 E)	$\mathtt{S}_1$:	*474* (51 min, 19 TO, 1 E)
	$\mathtt{S}_2$:	*355* (15 min, 8 TO)	$\mathtt{S}_2$:	-
$\mathcal{A}$ is max/plus	$\mathtt{W}$:	-	$\mathtt{W}$:	*216* (19 min, 7 TO, 1 E)
	$\mathtt{G}_1$:	*411* (25 min, 6 TO, 5 E)	$\mathtt{G}_1$:	*414* (58 min, 20 TO, 6 E)
	$\mathtt{G}_2$:	-	$\mathtt{G}_2$:	*235* (39 min, 12 TO)
	$\mathtt{S}_1$:	*467* (61 min, 18 TO, 1 E)	$\mathtt{S}_1$:	*424* (92 min, 34 TO, 6 E)
	$\mathtt{S}_2$:	-	$\mathtt{S}_2$:	*383* (40 min, 11 TO, 2 E)

methods automatically fail on many in the database. In contrast, this is not the case for $\mathtt{S}_1$ and $\mathtt{S}_2$, since they do not demand weak simplicity. Moreover, we can see that combination of different kinds of algebras improves termination proving power of $\mathtt{G}_1$ and $\mathtt{S}_1$, which is not possible for $\mathtt{W}$. We note that the aforementioned theoretical inclusions $\mathtt{W} \subsetneq X \subsetneq \mathtt{S}_1$ do not always hold, due to time-limit.

Comparison to the preliminary paper [87]. As explained, construction $\mathtt{S}_2$ corresponds to the generalization of the WPO proposed in the preliminary paper of this chapter. The results show that using another algebra yields a more powerful generalization of the WPO, namely $\mathtt{S}_1$.

3.6 Related Work

We conclude with related work.

Stronger AC-compatible orders. For termination modulo AC, a more powerful monotonic semantic path order [24] is known. (We remark that a flaw for monotonicity of the order is pointed out and corrected by [84].) An interesting line of research is to investigate a simulation result between the AC monotonic semantic path order and stronger AC-KBOs [64, 118]. This might pave a way to a more powerful variant of AC-GWPO.

General path order. For recursive comparison of arguments in path orders, we have used lexicographic extension in plain rewriting, and multiset extension in AC rewriting. It is well-known that path orders (e.g., recursive path orders) can be extended to allow choice of argument comparison method depending on

the head function symbols. General path orders (GPOs) [35, 44] are a unifying framework for such extensions, parameterizing the way to compare arguments. It is worth investigating simulation results between GPOs and suitable versions of WPOs.

Chapter 4

Generalized Weighted Path Order as Reduction Pair

We have considered the reduction order versions of the generalized weighted path order so far. On the other hand, the original weighted path order has been extended to reduction pairs (Theorem 2.7.8) so it can be used more effectively in the dependency pair method. In this chapter we introduce reduction pair versions of the generalized weighted path order, and ones of the semantic path order as their special cases. Interestingly, the result (Theorem 3.2.5) that the semantic path order as reduction order simulates the generalized weighted path order does *not* carry over to this reduction pair setting.

4.1 GWPO as Reduction Pair — Basic Version

We begin with a necessary definition. For a partial status π, a relation $\rightsquigarrow$ on terms is *simple with respect to* π if $f(t_1, \ldots, t_n) \rightsquigarrow t_i$ for all function symbols $f^{(n)}$, terms $t_1, \ldots, t_n$ and $i \in \pi(f)$. This is an analog of simplicity of algebras with respect to partial statuses which is demanded for the reduction pair version of WPO. As we expect, $>_{\mathcal{A}}$ (resp. $\geqslant_{\mathcal{A}}$) is simple with respect to π if an algebra $\mathcal{A}$ is strictly (resp. weakly) simple with respect to π.

Definition 4.1.1. *Let π be a partial status, and let $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ be pairs of relations on terms. The* generalized weighted path order *$(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ is a pair of relations on terms defined simultaneously as follows: $s \geqslant_{\mathsf{G}} t$ if*

1. *$s \sqsupset_A t$ or*
2. *$s \sqsupseteq_A t$ and one of the following conditions holds.*
   a. *$s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{G}} t$ for some $i \in \pi(f)$.*
   b. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{G}} t_j$ for all $j \in \pi(g)$, and*
      i. *$s \sqsupset_B t$ or*
      ii. *$s \sqsupseteq_B t$ and $\pi(f)(s_1, \ldots, s_m) \geqslant_{\mathsf{G}}^{\mathsf{lex}} \pi(g)(t_1, \ldots, t_n)$.*

c. $s \in \mathcal{V}$ *and* $s = t$

The relation $>_{\mathsf{G}}$ *is defined by cases 1, 2a and 2b where* $\geqslant_{\mathsf{G}}^{\mathsf{lex}}$ *is replaced by* $>_{\mathsf{G}}^{\mathsf{lex}}$*. Here* $(\geqslant_{\mathsf{G}}^{\mathsf{lex}}, >_{\mathsf{G}}^{\mathsf{lex}})$ *is the lexicographic extension of* $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$*.*

As in the reduction order setting, we define a monotonic version of the generalized weighted path order.

Definition 4.1.2. *Let* π *be a partial status, and let* $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ *and* $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ *be triples of relations on terms. The* monotonic generalized weighted path order $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ *is defined as the pair of* $\geqslant_A \cap \geqslant_B \cap \geqslant_{\mathsf{G}}$ *and* $\geqslant_A \cap \geqslant_B \cap >_{\mathsf{G}}$ *where* $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ *is induced by* $(\sqsupseteq_A, \sqsupset_A)$ *and* $(\sqsupseteq_B, \sqsupset_B)$*.*

Needless to say, the MGWPO as well as the GWPO are incremental (as reduction pair). Here we arrive at our main theorem.

Theorem 4.1.3. *Suppose that the signature is bounded. Let* π *be a partial status, and* $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ *and* $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ *reduction triples. If* $\sqsupseteq_A$ *is simple with respect to* π*, then* $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ *is a reduction pair and* $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ *is a normal reduction pair. If in addition* π *is total, then* $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ *is a monotone reduction pair.*

The proof appears in the end of this section. The original weighted path order as reduction pair (Theorem 2.7.8) is obtained by instantiating $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ with $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$, and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ with the reduction pair induced by precedence. As the original weighted path order does in [117, Section 4.2], it is possible to incorporate more refinements in Definition 4.1.1. This is considered in Section 4.3. Now, let us look at a termination proof by Theorem 4.1.3.

Example 4.1.4. *Let* $\mathcal{R}$ *be the following TRS (from [4, Example 3.4])*

$$\begin{aligned}
\mathsf{m}(x,0) &\xrightarrow{1} 0 & \mathsf{m}(\mathsf{s}(x),\mathsf{s}(y)) &\xrightarrow{2} \mathsf{m}(x,y)\\
\mathsf{p}(0,y) &\xrightarrow{3} y & \mathsf{p}(\mathsf{s}(x),y) &\xrightarrow{4} \mathsf{s}(\mathsf{p}(x,y))\\
& & \mathsf{m}(\mathsf{m}(x,y),z) &\xrightarrow{5} \mathsf{m}(x,\mathsf{p}(y,z))\\
\mathsf{q}(0,\mathsf{s}(y)) &\xrightarrow{6} 0 & \mathsf{q}(\mathsf{s}(x),\mathsf{s}(y)) &\xrightarrow{7} \mathsf{s}(\mathsf{q}(\mathsf{m}(x,y),\mathsf{s}(y)))
\end{aligned}$$

where m *means minus,* p *plus, and* q *quotient. The dependency pairs* $\mathsf{DP}(\mathcal{R})$ *consists of 6 rules.*

$$\begin{aligned}
\mathsf{M}(\mathsf{s}(x),\mathsf{s}(y)) &\xrightarrow{8} \mathsf{M}(x,y) & \mathsf{P}(\mathsf{s}(x),y) &\xrightarrow{9} \mathsf{P}(x,y)\\
\mathsf{M}(\mathsf{m}(x,y),z) &\xrightarrow{10} \mathsf{M}(x,\mathsf{p}(y,z)) & \mathsf{M}(\mathsf{m}(x,y),z) &\xrightarrow{11} \mathsf{P}(y,z)\\
\mathsf{Q}(\mathsf{s}(x),\mathsf{s}(y)) &\xrightarrow{12} \mathsf{Q}(\mathsf{m}(x,y),\mathsf{s}(y)) & \mathsf{Q}(\mathsf{s}(x),\mathsf{s}(y)) &\xrightarrow{13} \mathsf{M}(x,y)
\end{aligned}$$

Let $\mathcal{A}$ be the linear polynomial interpretation over $\mathbb{N}$

$$\begin{array}{llll} 0_{\mathcal{A}} = 0 & 0^{\sharp}_{\mathcal{A}} = 0 & \mathsf{s}_{\mathcal{A}}(x) = x+1 & \mathsf{s}^{\sharp}_{\mathcal{A}}(x) = 0 \\ \mathsf{m}_{\mathcal{A}}(x,y) = x & \mathsf{m}^{\sharp}_{\mathcal{A}}(x,y) = x & \mathsf{M}_{\mathcal{A}}(x,y) = x+1 & \mathsf{M}^{\sharp}_{\mathcal{A}}(x,y) = x \\ \mathsf{p}_{\mathcal{A}}(x,y) = x+y+1 & \mathsf{p}^{\sharp}_{\mathcal{A}}(x,y) = 1 & \mathsf{P}_{\mathcal{A}}(x,y) = 0 & \mathsf{P}^{\sharp}_{\mathcal{A}}(x,y) = x \\ \mathsf{q}_{\mathcal{A}}(x,y) = x+1 & \mathsf{q}^{\sharp}_{\mathcal{A}}(x,y) = 1 & \mathsf{Q}_{\mathcal{A}}(x,y) = x & \mathsf{Q}^{\sharp}_{\mathcal{A}}(x,y) = x \end{array}$$

and π the partial status with $\pi(\mathsf{M}) = \pi(\mathsf{m}) = [1]$ and $\pi(f) = [\,]$ for other function symbols f. We construct the reduction pair $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ by Theorem 4.1.3 with the reduction triples $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\geqslant_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}})$. To conclude that $\mathcal{R}$ is terminating, it suffices to show $\mathcal{R} \subseteq \geqslant_{\mathsf{MG}}$ and $\mathsf{DP}(\mathcal{R}) \subseteq >_{\mathsf{G}}$. This can be seen as follows: $\{1,2,3,6,8,11,12,13\} \subseteq >_{\mathcal{A}}$, so these are handled by case 1. For the other rules we compare them after marking the root symbols.

$$\begin{array}{lll} 4\colon & \mathsf{p}^{\sharp}(\mathsf{s}(x),y) >_{\mathcal{A}} \mathsf{s}^{\sharp}(\mathsf{p}(x,y)) & 5\colon \quad \mathsf{m}^{\sharp}(\mathsf{m}(x,y),z) \geqslant_{\mathcal{A}} \mathsf{m}^{\sharp}(x,\mathsf{p}(y,z)) \\ 7\colon & \mathsf{q}^{\sharp}(\mathsf{s}(x),\mathsf{s}(y)) >_{\mathcal{A}} \mathsf{s}^{\sharp}(\mathsf{q}(\mathsf{m}(x,y),\mathsf{s}(y))) & 9\colon \quad \mathsf{P}^{\sharp}(\mathsf{s}(x),y) >_{\mathcal{A}} \mathsf{P}^{\sharp}(x,y) \\ & & 10\colon \quad \mathsf{M}^{\sharp}(\mathsf{m}(x,y),z) \geqslant_{\mathcal{A}} \mathsf{M}(x,\mathsf{p}(y,z)) \end{array}$$

For rules 5 and 10 we further apply recursive comparison by case 2b(ii).

In the example above, some arguments ignored by π are used by the interpretation $\mathcal{A}$, see for example $\mathsf{p}_{\mathcal{A}}(x,y) = x+y+1$. This is an advantage of having partial status built-in rather than applying the GWPO with argument filtering. Although the two approaches are incomparable in general (as argument filtering allows collapsing), the following proposition states that the latter with a partial status (non-collapsing argument filter) can be simulated by the former. This is an analog of [117, Proposition 1].

Proposition 4.1.5. *Let $\mathcal{F}$ be the signature, π a partial status, and $(\geqslant_{A_1}, \sqsupseteq_{A_1}, \sqsupset_{A_1})$ and $(\geqslant_{B_1}, \sqsupseteq_{B_1}, \sqsupset_{B_1})$ reduction triples on $\mathcal{T}(\mathcal{F}^{\pi}, \mathcal{V})$. Then there are reduction triples $(\geqslant_{A_2}, \sqsupseteq_{A_2}, \sqsupset_{A_2})$ and $(\geqslant_{B_2}, \sqsupseteq_{B_2}, \sqsupset_{B_2})$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ such that $(\geqslant^{\pi}_{\mathsf{MG}_1}, >^{\pi}_{\mathsf{G}_1})$ is identical to $(\geqslant_{\mathsf{MG}_2}, >_{\mathsf{G}_2})$, where $(\geqslant_{\mathsf{MG}_1}, >_{\mathsf{G}_1})$ is induced by the total status, A_1 and B_1, and $(\geqslant_{\mathsf{MG}_2}, >_{\mathsf{G}_2})$ is induced by π, A_2 and B_2. Moreover, if $\sqsupseteq_{A_1}$ has the subterm property, then $\sqsupseteq_{A_2}$ is simple with respect to π.*

Proof. Define $(\geqslant_{A_2}, \sqsupseteq_{A_2}, \sqsupset_{A_2})$ as $(\geqslant^{\pi}_{A_1}, \sqsupseteq^{\pi}_{A_1}, \sqsupset^{\pi}_{A_1})$, and similarly $(\geqslant_{B_2}, \sqsupseteq_{B_2}, \sqsupset_{B_2})$ as $(\geqslant^{\pi}_{B_1}, \sqsupseteq^{\pi}_{B_1}, \sqsupset^{\pi}_{B_1})$. Then it can be shown that $s \geqslant^{\pi}_{\mathsf{G}_1} t$ if and only if $s \geqslant_{\mathsf{G}_2} t$ and $s >^{\pi}_{\mathsf{G}_1} t$ if and only if $s >_{\mathsf{G}_2} t$ for all terms s and t. This is done by induction on $|s| + |t|$. The simplicity of $\sqsupseteq_{A_2}$ is also easy. ∎

As another small remark, we compare the reduction pairs $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ and $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$. An evident difference is that the latter may be used as monotone

reduction pair with a total status. Another difference is that, in contrast to $(\gtrsim_{\mathsf{MG}}, >_{\mathsf{MG}})$, the reduction pair $(\gtrsim_{\mathsf{MG}}, >_{\mathsf{G}})$ may not satisfy normality (see the example below), which is a relevant property for the lexicographic combination technique [88]. As $>_{\mathsf{MG}} \subseteq >_{\mathsf{G}}$ holds by definition, we should use the former when the differences above are negligible.

Example 4.1.6. *Let $\mathcal{A}$ be the algebra on $\mathbb{N}$ defined by $\mathsf{f}_{\mathcal{A}}(x) = 0$ and $\mathsf{a}_{\mathcal{A}} = 1$. Let $(\gtrsim_{\mathsf{MG}}, >_{\mathsf{G}})$ be the GWPO induced by the empty status, $(\gtrsim_{\mathcal{A}}, \gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$ and the trivial reduction triple (Proposition 2.5.16). While $\mathsf{f}(\mathsf{a}) >_{\mathsf{G}} \mathsf{a}$ holds from case 1, $\mathsf{f}(\mathsf{a}) \gtrsim_{\mathsf{MG}} \mathsf{a}$ does not hold since we do not have $\mathsf{f}(\mathsf{a}) \gtrsim_{\mathcal{A}} \mathsf{a}$. So the reduction pair is not normal.*

As yet another remark, we relate the reduction pair version and the reduction order version introduced in Chapter 3: Let $>$ be the reduction order version of the MGWPO (Definition 3.1.3) induced by reduction triples A and B, and let $(\gtrsim_{\mathsf{MG}}, >_{\mathsf{MG}})$ be the reduction pair version of the MGWPO (Definition 4.1.2) induced by the total status, A and B. Then by easy simultaneous induction one can show that $\gtrsim \subseteq \gtrsim_{\mathsf{MG}}$ and $> \subseteq >_{\mathsf{MG}}$, where $\gtrsim$ is the reflexive closure of $>$. For termination proving, the greater relations are the more profitable, so we should use the reduction pair version.

Now, we present a proof of Theorem 4.1.3. Not surprisingly, proofs for the original WPO [114, 117] carry over. Let $(\gtrsim_{\mathsf{G}}, >_{\mathsf{G}})$ be the GWPO induced by a partial status π and pairs $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ of relations on terms.

Lemma 4.1.7. *The inclusion $>_{\mathsf{G}} \subseteq \gtrsim_{\mathsf{G}}$ holds.*

Proof. It suffices to prove that $s \gtrsim_{\mathsf{G}} t$ whenever $s >_{\mathsf{G}} t$ by induction on $|s| + |t|$, which is routine work. ∎

Lemma 4.1.8. *If $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are order pairs, then $\gtrsim_{\mathsf{G}}$ and $>_{\mathsf{G}}$ are transitive, and $(\gtrsim_{\mathsf{G}}, >_{\mathsf{G}})$ is compatible.*

Proof. The claim is proven by showing the next four properties for all terms s, t and u: if $s \gtrsim_{\mathsf{G}} t$ and $t \gtrsim_{\mathsf{G}} u$ then $s \gtrsim_{\mathsf{G}} u$; if $s \gtrsim_{\mathsf{G}} t$ and $t >_{\mathsf{G}} u$ then $s >_{\mathsf{G}} u$; if $s >_{\mathsf{G}} t$ and $t \gtrsim_{\mathsf{G}} u$ then $s >_{\mathsf{G}} u$; and if $s >_{\mathsf{G}} t$ and $t >_{\mathsf{G}} u$ then $s >_{\mathsf{G}} u$. This is done by easy simultaneous induction on $|s| + |t| + |u|$. ∎

Lemma 4.1.9. *If $\sqsupseteq_A$ and $\sqsupseteq_B$ are reflexive and $\sqsupseteq_A$ is simple with respect to π, then*

- *$t >_{\mathsf{G}} t_i$ whenever $i \in \pi(f)$ and $t = f(t_1, \ldots, t_n)$, and*
- *$t \gtrsim_{\mathsf{G}} t$ for all t (i.e., $\gtrsim_{\mathsf{G}}$ is reflexive).*

Proof. We show the two claims simultaneously by induction on $|t|$. For the first claim, we use the induction hypothesis to have $t_i \gtrsim_{\mathsf{G}} t_i$. Since $t \sqsupseteq_A t_i$ by assumption, we obtain $t >_{\mathsf{G}} t_i$ by case 2a. For the second claim, if t is a variable,

then $t \geqslant_{\mathsf{G}} t$ follows from case 2c and the assumptions. If $t = f(t_1, \ldots, t_n)$ we apply case 2b(ii) to obtain $t \geqslant_{\mathsf{G}} t$ as follows: we have $t \sqsupseteq_A t$ and $t \sqsupseteq_B t$ by assumption, and $t >_{\mathsf{G}} t_i$ for all $i \in \pi(f)$ and $\pi(f)(t_1, \ldots, t_n) \geqslant^{\mathsf{lex}} \pi(f)(t_1, \ldots, t_n)$ by the induction hypothesis. ∎

Next we show stability via Lemma 4.1.9.

Lemma 4.1.10. *If $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are stable, $\sqsupseteq_A$ and $\sqsupseteq_B$ are reflexive, and $\sqsupseteq_A$ is simple with respect to π, then $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ is stable.*

Proof. It suffices to prove the following properties: for all terms s and t and substitutions σ, if $s \geqslant_{\mathsf{G}} t$ then $s\sigma \geqslant_{\mathsf{G}} t\sigma$, and if $s >_{\mathsf{G}} t$ then $s\sigma >_{\mathsf{G}} t\sigma$. We prove these simultaneously by induction on $|s| + |t|$. Here we only prove the first property since the other can be proved in a similar way. To this end, we analyze how $s \geqslant_{\mathsf{G}} t$ is derived.

1. If $s \geqslant_{\mathsf{G}} t$ follows from case 1, then $s\sigma \sqsupset_A t\sigma$, so $s\sigma \geqslant_{\mathsf{G}} t\sigma$ by case 1.

2. If $s \geqslant_{\mathsf{G}} t$ follows from case 2, we have $s\sigma \sqsupseteq_A t\sigma$.
   a. Suppose that $s \geqslant_{\mathsf{G}} t$ follows from case 2a. By the induction hypothesis, we obtain $s\sigma \geqslant_{\mathsf{G}} t\sigma$ again by case 2a.
   b. Suppose that $s \geqslant_{\mathsf{G}} t$ follows from case 2b. Whether it is case 2b(i) or 2b(ii), with the induction hypothesis, we obtain $s\sigma \geqslant_{\mathsf{G}} t\sigma$ again by case 2b.
   c. Suppose that $s \geqslant_{\mathsf{G}} t$ follows from case 2c, namely $s = t$. Then $s\sigma \geqslant_{\mathsf{G}} t\sigma$ follows from Lemma 4.1.9. ∎

We prove well-foundedness via Buchholz's method, cf. [117, Lemma 9]. We need the assumptions to use Lemma 4.1.8.

Lemma 4.1.11. *Suppose that the signature is bounded, $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are well-founded stable order pairs, and $\sqsupseteq_A$ is simple with respect to π. Then $t \in \mathsf{SN}(>_{\mathsf{G}})$ whenever $s = f(s_1, \ldots, s_m) >_{\mathsf{G}} t$ and $s_i \in \mathsf{SN}(>_{\mathsf{G}})$ for all $i \in \pi(f)$.*

Proof. We perform induction on the quadruple $(s, s, \pi(f)(s_1, \ldots, s_m), t)$ with respect to $\gg$, where $\gg$ is the lexicographic product

$$(\sqsupseteq_A, \sqsupset_A) \otimes (\sqsupseteq_B, \sqsupset_B) \otimes (\geqslant_{\mathsf{G}}^{\mathsf{lex}}, >_{\mathsf{G}}^{\mathsf{lex}}) \otimes (\trianglerighteq, \triangleright)$$

defined on $\mathcal{T} \times \mathcal{T} \times \mathsf{SN}(>_{\mathsf{G}})^{\leqslant k} \times \mathcal{T}$ for an upper bound k of arity. It suffices to prove that $u \in \mathsf{SN}(>_{\mathsf{G}})$ whenever $t >_{\mathsf{G}} u$. If t is a variable, then by definition $t \sqsupset_A u$ must hold. By applying $\sigma = \{t \mapsto u\}$ to $t \sqsupset_A u$ successively, we obtain $u \sqsupset_A u\sigma \sqsupset_A (u\sigma)\sigma \sqsupset_A \cdots$, which contradicts to the well-foundedness of $\sqsupset_A$. So letting $t = g(t_1, \ldots, t_m)$ we analyze the derivation of $s >_{\mathsf{G}} t$.

1. Suppose $s \sqsupset_A t$. By the simplicity assumption, we have $s \sqsupset_A t \sqsupseteq_A t_j$ and therefore $s >_{\mathsf{G}} t_j$ for all $j \in \pi(g)$. So, we apply the induction hypothesis with $(s, s, \pi(f)(s_1, \ldots, s_m), \underline{t}) \gg (s, s, \pi(f)(s_1, \ldots, s_m), \underline{t_j})$ to obtain $t_j \in \mathsf{SN}(>_{\mathsf{G}})$ for all $j \in \pi(g)$. Here, for readability, the decreasing components are indicated by underlining. Then, we apply the induction hypothesis with $(\underline{s}, s, \pi(f)(s_1, \ldots, s_m), t) \gg (\underline{t}, t, \pi(g)(t_1, \ldots, t_n), u)$ to obtain $u \in \mathsf{SN}(>_{\mathsf{G}})$.

2. Suppose $s \sqsupseteq_A t$.
   a. Assume $s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{G}} t$ for some $i \in \pi(f)$. Then we have $s_i >_{\mathsf{G}} u$ by Lemma 4.1.8, so $s_i \in \mathsf{SN}(>_{\mathsf{G}})$ yields $u \in \mathsf{SN}(>_{\mathsf{G}})$.
   b. Assume $s >_{\mathsf{G}} t_j$ for all $j \in \pi(g)$. We apply the induction hypothesis with $(s, s, \pi(f)(s_1, \ldots, s_m), \underline{t}) \gg (s, s, \pi(f)(s_1, \ldots, s_m), \underline{t_j})$ to obtain $t_j \in \mathsf{SN}(>_{\mathsf{G}})$ for all $j \in \pi(g)$. Finally, we apply the induction hypothesis with

      $$(s, s, \pi(f)(s_1, \ldots, s_m), t) \gg (t, t, \pi(g)(t_1, \ldots, t_n), u)$$

      to obtain $u \in \mathsf{SN}(>_{\mathsf{G}})$, where the inequality of $\gg$ is justified as follows: if case 2b(i) holds, then we have a decrease in the second component; if case 2b(ii) holds, then we have a decrease in the third. ∎

Lemma 4.1.12. *Suppose that the signature is bounded, $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are well-founded stable order pairs, and $\sqsupseteq_A$ is simple with respect to π Then $>_{\mathsf{G}}$ is well-founded.*

Proof. We show that $t \in \mathsf{SN}(>_{\mathsf{G}})$ for all terms t by structural induction on t. It suffices to show that $u \in \mathsf{SN}(>_{\mathsf{G}})$ whenever $t >_{\mathsf{G}} u$. By the same argument used in the proof of Lemma 4.1.11, we can assume that t is not a variable, so let $t = f(t_1, \ldots, t_m)$. By the induction hypothesis $t_i \in \mathsf{SN}(>_{\mathsf{G}})$ for all i, so Lemma 4.1.11 applies. ∎

Finally, we prove monotonicity. Let $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ be the monotone generalized weighted path order induced by a partial status π and triples $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ of relations on terms.

Lemma 4.1.13. *Suppose that $\geqslant_A$ and $\geqslant_B$ are monotone, $\sqsupseteq_A$ and $\sqsupseteq_B$ are reflexive and harmonious to $\geqslant_A$ and $\geqslant_B$, respectively, and $\sqsupseteq_A$ is simple with respect to π. Then $\geqslant_{\mathsf{MG}}$ is monotone. If in addition π is total, then $>_{\mathsf{MG}}$ is also monotone.*

Proof. The claim can be shown with Lemma 4.1.9. ∎

Theorem 4.1.3 is now obtained by invoking these lemmas.

The usable rule technique [46–48, 54, 105] is a popular refinement to Theorem 2.7.3 which relaxes the requirement $\mathcal{R} \subseteq \geqslant$ to $\mathcal{U} \subseteq \geqslant$ for a possibly smaller set $\mathcal{U} \subseteq \mathcal{R}$ of usable rules; see also [92, Section 4] for the most general version.

The technique demands the reduction pair $(\geqslant, >)$ to be $\mathcal{C}_\mathcal{E}$*-compatible*, meaning that the reduction pair $(\geqslant, >)$ can be extended to one $(\geqslant', >')$ with $\mathcal{C}_\mathcal{E} \subseteq \geqslant'$, where $\mathcal{C}_\mathcal{E} = \{\mathsf{c}(x,y) \to x, \mathsf{c}(x,y) \to y\}$ and c is a fresh binary symbol. Like usual reduction pairs, the GWPO (Theorem 4.1.3) satisfies $\mathcal{C}_\mathcal{E}$-compatibility almost for free, as we show in what follows.

We say that a reduction triple $(\geqslant, \sqsupseteq, \sqsupset)$ is $\mathcal{C}_\mathcal{E}$*-compatible* if it can be extended to one $(\geqslant', \sqsupseteq', \sqsupset')$ with $\mathcal{C}_\mathcal{E} \subseteq \geqslant'$.

Proposition 4.1.14. *Let A and B be $\mathcal{C}_\mathcal{E}$-compatible reduction triples and π a partial status. Then $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ and $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ are $\mathcal{C}_\mathcal{E}$-compatible.*

Proof. The claim is shown by extending π with $\pi(\mathsf{c}) = [1,2]$. ∎

The assumption of the proposition above is negligible in practice, because usual reduction triples are $\mathcal{C}_\mathcal{E}$-compatible: the trivial reduction triple (Proposition 2.5.16) and those induced by precedences (Proposition 2.5.11) are $\mathcal{C}_\mathcal{E}$-compatible; usual interpretations (like polynomial interpretations, max/plus interpretations, matrix interpretation [37], and so on) induce $\mathcal{C}_\mathcal{E}$-compatible reduction triples, whether via Proposition 2.5.12 or Proposition 2.5.14; and lexicographic combination (Proposition 2.5.15) preserves $\mathcal{C}_\mathcal{E}$-compatibility.

Continuing with the usable rule technique, we consider π*-compatibility* of reduction pairs, which is relevant to the refinement called usable rule w.r.t. argument filtering [48]. Let us recall the formal definition: The i-th argument position of a function symbol f is *invariant* w.r.t. a relation $\rightsquigarrow$ if $C[s] \rightsquigarrow C[t]$ for all terms s and t and contexts $C = f(\bar{u}, \Box, \bar{v})$ with the hole $\Box$ at the i-th argument of f [88]. A reduction pair $(\geqslant, >)$ is π-compatible for a partial status π if the i-th argument position of f is invariant w.r.t. $\geqslant$ whenever $i \notin \pi(f)$. So, below we give a sufficient condition of invariance w.r.t. $\geqslant_{\mathsf{MG}}$. For convenience, we extend the notion of invariance to reduction triples: The i-th argument position of a function symbol f is invariant w.r.t. a reduction triple $(\geqslant, \sqsupseteq, \sqsupset)$ if it is so with respect to $\geqslant$ and $\sqsupseteq$.

Proposition 4.1.15. *Let $\geqslant_{\mathsf{MG}}$ be the MGWPO induced by a partial status π and reduction triples A and B. The i-th argument position of a function symbol f is invariant w.r.t. $\geqslant_{\mathsf{MG}}$ if it is so w.r.t. A and B, and also $i \notin \pi(f)$.*

Proof. Let s and t be terms and $C = f(\bar{u}, \Box, \bar{v})$ a context with the hole $\Box$ at the i-th argument of f. By assumption, we have $C[s] \geqslant_A C[t]$, $C[s] \geqslant_B C[t]$ and $C[s] \geqslant_{\mathsf{G}} C[t]$ by case 2b(ii). So $C[s] \geqslant_{\mathsf{MG}} C[t]$. ∎

We can use the criterion above as follows: Given a partial status π and reduction triples A and B, we define the partial statuses ρ by

$$\rho(f) = \{i \notin \pi(f) \mid \text{the } i\text{-th argument position of } f \text{ is invariant w.r.t. } A \text{ and } B\}$$

for each function symbols f. Also, we define $\bar{\rho}(f) = \{1, \ldots, n\} \setminus \rho(f)$ for each n-ary function symbol f. Then $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ and $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ induced by π, A and B are $\bar{\rho}$-compatible.

The remaining problem for using Proposition 4.1.15 is how to estimate invariant positions for the underlying reduction triples A and B. For those induced by algebras, this is done by adapting how to do it for reduction pairs, see for example [88]. For the trivial reduction triple and those induced by precedences, all argument positions are invariant. Also, lexicographic combination (Proposition 2.5.15) preserves invariance: if an argument position is invariant w.r.t. reduction triples A and B, then it is so w.r.t. the lexicographic combination of A and B.

Another technique to which invariance is relevant is lexicographic combination of reduction pairs [88]. In addition to invariant positions, the technique demands us to identify *monotone* positions: Formally, the i-th argument position of a function symbol f is monotone w.r.t. a relation $\rightsquigarrow$ if $f(t_1, \ldots, t_i, \ldots, t_n) \rightsquigarrow f(t_1, \ldots, t_i', \ldots, t_n)$ for all terms $t_1, \ldots, t_n$ and t_i with $t_i \rightsquigarrow t_i'$. We have a proposition for monotone positions in the style of Proposition 4.1.15.

Proposition 4.1.16. *Let $>_{\mathsf{MG}}$ the MGWPO induced by a partial status π and reduction triples A and B. The i-th argument position of a function symbol f is monotone w.r.t. $>_{\mathsf{MG}}$ if $i \in \pi(f)$.* ∎

It seems non-trivial to estimate monotone positions of $>_{\mathsf{G}}$, because it is not designed to be monotone in the first place. That said, the lexicographic combination technique demands normality, and therefore forces to use $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ instead of $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$.

4.2 SPO as Reduction Pair — Basic Version

To the author's best knowledge, there is no existing work to formulate the semantic path order as reduction pair incorporating mutual recursion and partial status as in Definition 4.1.1. Such a version of the semantic path order is naturally obtained from the GWPO.

Definition 4.2.1. *Let π be a partial status and let $(\sqsupseteq, \sqsupset)$ be a pair of relations on terms. The* semantic path order *$(\geqslant_{\mathsf{S}}, >_{\mathsf{S}})$ induced by π and $(\sqsupseteq, \sqsupset)$ is defined inductively as follows: $s \geqslant_{\mathsf{S}} t$ if*

1. *$s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{S}} t$ for some $i \in \pi(f)$.*
2. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{S}} t_j$ for all $j \in \pi(g)$, and*
   a. *$s \sqsupset t$ or*

b. $s \sqsupseteq t$ *and* $\pi(f)(s_1,\dots,s_m) \geqslant_{\mathsf{S}}^{\mathsf{lex}} \pi(g)(t_1,\dots,t_n)$.

3. $s \in \mathcal{V}$ *and* $s = t$

The relation $>_{\mathsf{S}}$ *is defined by cases 1, 2a and 2b where* $\geqslant_{\mathsf{S}}^{\mathsf{lex}}$ *is replaced by* $>_{\mathsf{S}}^{\mathsf{lex}}$. *Here* $(\geqslant_{\mathsf{S}}^{\mathsf{lex}}, >_{\mathsf{S}}^{\mathsf{lex}})$ *is the lexicographic extension of* $(\geqslant_{\mathsf{S}}, >_{\mathsf{S}})$.

Obviously, we could define $(\geqslant_{\mathsf{S}}, >_{\mathsf{S}})$ as $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ induced from $(\mathcal{T}\times\mathcal{T}, \varnothing)$ and $(\sqsupseteq, \sqsupset)$. The monotone version is defined as follows.

Definition 4.2.2. *Let* π *be a partial status, and let* $(\geqslant, \sqsupseteq, \sqsupset)$ *be a triple of relations on terms. The* monotonic semantic path order $(\geqslant_{\mathsf{MS}}, >_{\mathsf{MS}})$ *is defined as the pair of* $\geqslant \cap \geqslant_{\mathsf{S}}$ *and* $\geqslant \cap >_{\mathsf{S}}$ *where* $(\geqslant_{\mathsf{S}}, >_{\mathsf{S}})$ *is induced by* $(\sqsupseteq, \sqsupset)$.

Again, we could define $(\geqslant_{\mathsf{MS}}, >_{\mathsf{MS}})$ as $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ induced from the reduction triple $(\mathcal{T}\times\mathcal{T}, \mathcal{T}\times\mathcal{T}, \varnothing)$ and $(\geqslant, \sqsupseteq, \sqsupset)$. This results in the following corollary.

Corollary 4.2.3. *Suppose that the signature is bounded. Let* π *be a partial status, and* $(\geqslant, \sqsupseteq, \sqsupset)$ *a reduction triple. Then* $(\geqslant_{\mathsf{MS}}, >_{\mathsf{S}})$ *is a reduction pair. If in addition* π *is total, then* $(\geqslant_{\mathsf{MS}}, >_{\mathsf{MS}})$ *is a monotone reduction pair.* ∎

The advantage compared to Theorem 4.1.3 is that the weak simplicity requirement is vanished.

Example 4.2.4. *Consider the following singleton TRS* $\mathcal{R}$ *[33].*

$$\mathsf{if}(\mathsf{if}(x,y,z),v,w) \xrightarrow{1} \mathsf{if}(x,\mathsf{if}(y,v,w),\mathsf{if}(z,v,w))$$

We show the termination by Corollary 2.7.4 by constructing an MSPO $(\geqslant_{\mathsf{MS}}, >_{\mathsf{S}})$ *such that* $\mathcal{R} \subseteq \geqslant_{\mathsf{MS}}$ *and* $\mathsf{DP}(\mathcal{R}) \subseteq >_{\mathsf{S}}$. *Here the dependency pairs* $\mathsf{DP}(\mathcal{R})$ *are:*

$$\begin{aligned}
\mathsf{IF}(\mathsf{if}(x,y,z),v,w) &\xrightarrow{2} \mathsf{IF}(x,\mathsf{if}(y,v,w),\mathsf{if}(z,v,w))\\
\mathsf{IF}(\mathsf{if}(x,y,z),v,w) &\xrightarrow{3} \mathsf{IF}(y,v,w)\\
\mathsf{IF}(\mathsf{if}(x,y,z),v,w) &\xrightarrow{4} \mathsf{IF}(z,v,w)
\end{aligned}$$

To that end, consider $(\geqslant_{\mathsf{MS}}, >_{\mathsf{S}})$ *induced by* π *and* $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$, *where* $\mathcal{A}$ *is the linear polynomial interpretation over* $\mathbb{N}$

$$\begin{aligned}
\mathsf{if}_{\mathcal{A}}(x,y,z) &= x+y+1 & \qquad \mathsf{if}^{\sharp}_{\mathcal{A}}(x,y,z) &= 0\\
\mathsf{IF}_{\mathcal{A}}(x,y,z) &= x+y+z & \qquad \mathsf{IF}^{\sharp}_{\mathcal{A}}(x,y,z) &= 0
\end{aligned}$$

and π *is the partial status with* $\pi(\mathsf{if}) = [1,2,3]$ *and* $\pi(\mathsf{IF}) = [1]$. *By calculation, both terms of* $\mathsf{if}(\mathsf{if}(x,y,z),v,w) \geqslant_{\mathcal{A}} \mathsf{if}(x,\mathsf{if}(y,v,w),\mathsf{if}(z,v,w))$ *evaluate to* $x+y+v+2$,

so the inequality holds. In this way, we can confirm $\mathcal{R} \cup \mathsf{DP}(\mathcal{R}) \subseteq \gtrsim_{\mathcal{A}}$. *Now, it suffices to show* $\mathcal{R} \cup \mathsf{DP}(\mathcal{R}) \subseteq >_{\mathsf{S}}$. *Actually, we can use case 2b for all rules 1–4 by verifying that the first arguments are decreasing, thanks to the choice of* π. *So we conclude the termination of* $\mathcal{R}$, *remarking that the third argument of* $\mathsf{if}_{\mathcal{A}}$ *is not weakly simple but* $3 \in \pi(\mathsf{if})$.

Next, following Section 3.2 we compare the generalized weighted path order with the semantic path order as reduction pair. Let $(\gtrsim_{\mathsf{G}}, >_{\mathsf{G}})$ be the GWPO induced by a partial status π and stable order pairs $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ such that $\sqsupseteq_A$ is simple with respect to π. We consider the SPO $(\gtrsim_{\mathsf{S}}, >_{\mathsf{S}})$ induced by the lexicographic combination $(\sqsupseteq_{AB}, \sqsupset_{AB})$ and π. In this setting, an analog of Lemma 3.2.4 holds.

Lemma 4.2.5. *For all terms* s *and* t, *it holds that* $s \gtrsim_{\mathsf{S}} t$ *implies* $s \gtrsim_{\mathsf{G}} t$, *and* $s >_{\mathsf{S}} t$ *implies* $s >_{\mathsf{G}} t$.

Proof. We prove the two implications simultaneously by induction on $|s| + |t|$. Here we only prove the first claim, as the other is analogously shown. Let $s \gtrsim_{\mathsf{S}} t$. We analyze how it is derived.

1. If it follows from case 1, then $s = f(s_1, \ldots, s_m)$ and $s_i \gtrsim_{\mathsf{S}} t$ for some $i \in \pi(f)$. By the induction hypothesis we have $s_i \gtrsim_{\mathsf{G}} t$. By definition, $s_i \sqsupset_A t$ or $s_i \sqsupseteq_A t$ holds. In the former case, we use the simplicity assumption to obtain $s \sqsupseteq_A s_i \sqsupset_A t$, so $s \gtrsim_{\mathsf{G}} t$ by case 1. In the latter case, we are done by case 2a.

2. If it follows from case 2, we have $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$ and $s >_{\mathsf{S}} t_j$ for all $j \in \pi(g)$. By the induction hypothesis we have $s >_{\mathsf{G}} t_j$ for all $j \in \pi(g)$.
   a. If $s \sqsupset_{AB} t$, we further analyze how it follows: If $s \sqsupset_A t$ then we have $s >_{\mathsf{G}} t$ by case 1. Otherwise, we conclude with case 2a(i).
   b. If $s \sqsupseteq_{AB} t$, we further analyze how it follows: If $s \sqsupset_A t$ then we have $s >_{\mathsf{G}} t$ by case 1. Otherwise, we conclude with case 2a(ii) using the induction hypothesis.

3. If it follows from case 3, we simply apply the corresponding case 2c. ∎

The assumption that $\sqsupseteq_A$ is simple with respect to π is essential for the lemma above.

Example 4.2.6. *Consider the signature* $\{\mathsf{a}^{(0)}, \mathsf{f}^{(1)}\}$, *the partial status* π *with* $\pi(\mathsf{f}) = [1]$, *and the algebra* $\mathcal{A}$ *on* $\mathbb{N}$ *defined by* $\mathsf{a}_{\mathcal{A}} = 1$ *and* $\mathsf{f}_{\mathcal{A}}(x) = 0$. *Let* $(\sqsupseteq_A, \sqsupset_A)$ *be* $(\gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$ *and let* $(\sqsupseteq_B, \sqsupset_B)$ *be* $(\mathcal{T} \times \mathcal{T}, \varnothing)$. *Then we have* $\mathsf{f}(\mathsf{a}) >_{\mathsf{S}} \mathsf{a}$ *by case 1 but* $\mathsf{f}(\mathsf{a}) >_{\mathsf{G}} \mathsf{a}$ *does not hold.*

The converse is shown by additionally assuming that π is total. The proof follows the same line as Lemma 3.2.3.

Lemma 4.2.7. *Suppose that π is total. For all terms s and t, it holds that $s \geqslant_{\mathsf{G}} t$ implies $s \geqslant_{\mathsf{S}} t$, and $s >_{\mathsf{G}} t$ implies $s >_{\mathsf{S}} t$.* ∎

The assumption that π is total cannot be dropped.

Example 4.2.8. *Suppose that the signature only contains a unary function symbol f. Let π be the partial status defined by $\pi(\mathsf{f}) = [\,]$, let $\mathcal{A}$ be the algebra on $\mathbb{N}$ defined by $\mathsf{f}_{\mathcal{A}}(x) = x + 1$, and let $(\sqsupseteq_B, \sqsupset_B)$ be an arbitrary pair of relations on terms (e.g., one induced by a precedence to consider the original WPO). On one hand, the GWPO $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ induced by $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\sqsupseteq_B, \sqsupset_B)$ satisfies $\mathsf{f}(x) \geqslant_{\mathsf{G}} x$ and $\mathsf{f}(x) >_{\mathsf{G}} x$ by case 1. On the other hand, no SPO $(\geqslant_{\mathsf{S}}, >_{\mathsf{S}})$ (regardless of the underlying order pair) satisfies neither $\mathsf{f}(x) \geqslant_{\mathsf{S}} x$ nor $\mathsf{f}(x) >_{\mathsf{S}} x$, because only possible case 1 is blocked by $\pi(\mathsf{f}) = [\,]$.*

The cause behind the example above is very simple: if $s \sqsupset_A t$, the GWPO satisfies $s >_{\mathsf{G}} t$ and $s >_{\mathsf{G}} t$ with case 1 regardless of the underlying partial status, while $s >_{\mathsf{S}} t$ (or $s \geqslant_{\mathsf{S}} t$) may not be possible if the partial status does not allow it, cf. Example 3.2.2.

As a consequence of Lemmas 4.2.5 and 4.2.7, we can show an analog of Theorem 3.2.5: Let $(\geqslant_{\mathsf{MG}}, >_{\mathsf{MG}})$ be the monotone generalized weighted path order induced by a partial status π and reduction triples $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ with $\sqsupseteq_A$ simple with respect to π, and let $(\geqslant_{\mathsf{MS}}, >_{\mathsf{MS}})$ be the monotone semantic path order induced by π and $(\geqslant_A \cap \geqslant_B, \sqsupseteq_{AB}, \sqsupset_{AB})$.

Theorem 4.2.9. *We have $\geqslant_{\mathsf{MS}} \subseteq \geqslant_{\mathsf{MG}}$ and $>_{\mathsf{MS}} \subseteq >_{\mathsf{MG}}$ as well as $>_{\mathsf{S}} \subseteq >_{\mathsf{G}}$. If additionally π is total, then $\geqslant_{\mathsf{MS}}$, $>_{\mathsf{MS}}$ and $>_{\mathsf{S}}$ are identical to $\geqslant_{\mathsf{MG}}$, $>_{\mathsf{MG}}$ and $>_{\mathsf{G}}$, respectively.* ∎

So, in the setting of Theorem 4.2.9 the GWPO is preferred over the SPO if $\sqsupseteq_A$ is simple with respect to π. In other words, the SPO should be used only if the simplicity requirement is not satisfied.

Compared to Theorem 3.2.5, we have seen that the construction used in Theorem 4.2.9 it is good enough to simulate GWPOs with full statuses but not good enough to simulate *all* of them. So, it is natural to seek a better construction. However, we note that any such construction must essentially use well-foundedness of the order pair A, in contrast to Theorems 3.2.5 and 4.2.9. This can be seen from the following example.

Example 4.2.10. *Suppose that the signature only contains a unary function symbol f. Let π be the partial status defined by $\pi(\mathsf{f}) = [\,]$, let $\mathcal{A}$ be the algebra on $\mathbb{Z}_{\leqslant 0}$ defined by $\mathsf{f}_{\mathcal{A}}(x) = x - 1$, and let $(\sqsupseteq_B, \sqsupset_B)$ be an arbitrary pair of relations on terms. On one hand, the GWPO $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ induced by $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\sqsupseteq_B, \sqsupset_B)$ satisfies $x \geqslant_{\mathsf{G}} \mathsf{f}(x)$ and $x >_{\mathsf{G}} \mathsf{f}(x)$ by case 1 and $x >_{\mathcal{A}} \mathsf{f}(x)$. On the other hand, no SPO (regardless of the underlying order pair and partial status) satisfies it, because by definition the variable x cannot be greater than or equal to $\mathsf{f}(x)$.*

In Chapter 5 we indeed consider non-well-founded order pairs as above.

4.3 GWPO as Reduction Pair — Refined Version

As already mentioned, the original WPO has a version with more refinements. The definition is as follows:

Definition 4.3.1. *Let π be a partial status, $\mathcal{A}$ an algebra, and $\succeq$ a precedence (a quasi-order on function symbols). The* weighted path order *$(\geqslant_{\mathsf{W}'}, >_{\mathsf{W}'})$ is the pair of relations on terms defined simultaneously as follows: $s \geqslant_{\mathsf{W}'} t$ if*

1. *$s >_{\mathcal{A}} t$, or*
2. *$s \geqslant_{\mathcal{A}} t$ and one of the following conditions holds:*
   a. *$s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{W}'} t$ for some $i \in \pi(f)$.*
   b. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{W}'} t_j$ for all $j \in \pi(g)$, and*
      i. *$f \succ g$ or*
      ii. *$f \succeq g$ and $\pi(f)(s_1, \ldots, s_m) \geqslant^{\mathsf{lex}}_{\mathsf{W}'} \pi(g)(t_1, \ldots, t_n)$.*
   c. *$s \in \mathcal{V}$ and either $s = t$ or $t = g(t_1, \ldots, t_n)$, $\pi(g) = [\,]$ and g is least in $\succeq$.*
   d. *$s = f(s_1, \ldots, s_m)$, $t \in \mathcal{V}$, $\mathcal{A}$ is strictly simple with respect to π, and for all function symbols g, either $f \succ g$ or $f \equiv g$ and $\pi(g) = [\,]$ hold.*

Here $\succ$ is the strict part of $\succeq$ and $\equiv$ is the equivalence induced from $\succeq$. The relation $>_{\mathsf{W}'}$ is defined by cases 1, 2a and 2b where $\geqslant^{\mathsf{lex}}_{\mathsf{W}'}$ is replaced by $>^{\mathsf{lex}}_{\mathsf{W}'}$. As usual, $(\geqslant^{\mathsf{lex}}_{\mathsf{W}'}, >^{\mathsf{lex}}_{\mathsf{W}'})$ is the lexicographic extension of $(\geqslant_{\mathsf{W}'}, >_{\mathsf{W}'})$.

This refined version also gives rise to reduction pairs.

Theorem 4.3.2 ([117])**.** *Suppose that the signature is bounded. Let π be a partial status, $\succeq$ a precedence, and $\mathcal{A}$ an algebra that is well-founded, non-trivial, normal, weakly π-simple and weakly monotone. Then $(\geqslant_{\mathsf{W}'}, >_{\mathsf{W}'})$ is a reduction pair.*

Here, the normality requirement is required to keep $\geqslant_{\mathsf{W}'}$ monotone, see Example 2.4.9. The non-triviality of the algebra is required for $\geqslant_{\mathsf{G}'}$ to be transitive, see [103]. As a consequence, the refined version has the nice property that it can simulate every non-trivial and *normal* reduction pair.

Proposition 4.3.3 ([117])**.** *Let $(\geqslant, >)$ be a non-trivial and normal reduction pair. Then there are a partial status π, a precedence $\succeq$, and $\mathcal{A}$ an algebra that is well-founded, non-trivial, normal, weakly π-simple and weakly monotone such that the induced WPO $(\geqslant_{\mathsf{W}'}, >_{\mathsf{W}'})$ is identical to $(\geqslant, >)$.*

We incorporate the refinements into our generalized weighted path order.

Definition 4.3.4. *Let π be a partial status, and let $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ be pairs of relations on terms. The* generalized weighted path order *$(\gtrsim_{\mathsf{G}'}, >_{\mathsf{G}'})$ is a pair of relations on terms defined simultaneously as follows: $s \gtrsim_{\mathsf{G}'} t$ if*

1. *$s \sqsupset_A t$ or*
2. *$s \sqsupseteq_A t$ and one of the following conditions holds.*
   a. *$s = f(s_1, \ldots, s_m)$ and $s_i \gtrsim_{\mathsf{G}'} t$ for some $i \in \pi(f)$.*
   b. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{G}'} t_j$ for all $j \in \pi(g)$, and*
      i. *$s \sqsupset_B t$ or*
      ii. *$s \sqsupseteq_B t$ and $\pi(f)(s_1, \ldots, s_m) \gtrsim^{\mathsf{lex}}_{\mathsf{G}'} \pi(g)(t_1, \ldots, t_n)$.*
   c. *$s \in \mathcal{V}$ and either $s = t$ or $t = g(\bar{t})$, $\pi(g) = [\,]$ and $f(\bar{x}) \sqsupseteq_B t$ for all $f \in \mathcal{F}$ and variables $\bar{x}$.*
   d. *$s = f(\bar{s})$, $t \in \mathcal{V}$, $\sqsupset_A$ is simple with respect to π, and for all function symbols g and variables $\bar{x}$, it holds that $s \sqsupset_B g(\bar{x})$ or $s \sqsupseteq_B g(\bar{x})$ and $\pi(g) = [\,]$.*

The relation $>_{\mathsf{G}'}$ is defined by cases 1, 2a and 2b where $\gtrsim^{\mathsf{lex}}_{\mathsf{G}'}$ is replaced by $>^{\mathsf{lex}}_{\mathsf{G}'}$. Here $(\gtrsim^{\mathsf{lex}}_{\mathsf{G}'}, >^{\mathsf{lex}}_{\mathsf{G}'})$ is the lexicographic extension of $(\gtrsim_{\mathsf{G}'}, >_{\mathsf{G}'})$.

The monotonic version is defined in the same way.

Definition 4.3.5. *Let π be a partial status, and let $(\gtrsim_A, \sqsupseteq_A, \sqsupset_A)$ and $(\gtrsim_B, \sqsupseteq_B, \sqsupset_B)$ be triples of relations on terms. The* monotonic generalized weighted path order *$(\gtrsim_{\mathsf{MG}'}, >_{\mathsf{MG}'})$ is defined as the pair of $\gtrsim_A \cap \gtrsim_B \cap \gtrsim_{\mathsf{G}'}$ and $\gtrsim_A \cap \gtrsim_B \cap >_{\mathsf{G}'}$ where $(\gtrsim_{\mathsf{G}'}, >_{\mathsf{G}'})$ is induced by $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$.*

Again, this version of the (M)GWPO is also incremental.

Theorem 4.3.6. *Suppose that the signature is bounded. Let π be a partial status, and $(\gtrsim_A, \sqsupseteq_A, \sqsupset_A)$ and $(\gtrsim_B, \sqsupseteq_B, \sqsupset_B)$ reduction triples with $\sqsupseteq_A$ simple with respect to π. If $(\sqsupseteq_A, \sqsupset_A)$ is non-trivial, $(\gtrsim_{\mathsf{MG}'}, >_{\mathsf{G}'})$ is a reduction pair. If in addition π is total, $(\gtrsim_{\mathsf{MG}'}, >_{\mathsf{MG}'})$ is a monotone reduction pair.*

Before presenting the proof, we add several remarks. As expected, Theorem 4.3.2 is obtained from Theorem 4.3.6 by instantiating $(\gtrsim_A, \sqsupseteq_A, \sqsupset_A)$ with $(\gtrsim_{\mathcal{A}}, \gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$, and $(\gtrsim_B, \sqsupseteq_B, \sqsupset_B)$ with the reduction pair induced by precedence. We also note that for $(\gtrsim_{\mathsf{MG}'}, >_{\mathsf{G}'})$ to be a reduction pair demands non-triviality of $(\sqsupseteq_A, \sqsupset_A)$, which is already required in the refined version of the original WPO, in the form of non-triviality of the algebra $\mathcal{A}$. This is relevant when plugging the trivial reduction triple to obtain an SPO version, which is considered in the next section.

Moreover, the refined version of the GWPO has the nice property that it can simulate every non-trivial reduction pair.

Theorem 4.3.7. *Let $(\geqslant, >)$ be a non-trivial reduction pair. Then there are a partial status π and reduction triples $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ with $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$ simple with respect to π such that $(\geqslant_{\mathsf{MG}'}, >_{\mathsf{G}'})$ is identical to $(\geqslant, >)$.*

Proof. We follow the proof of Proposition 4.3.3 [117]. Let π be the empty status, and let A be $(\geqslant, \geqslant, >)$ and B the reduction triple induced by the smallest precedence. Then the same argument goes through. ∎

Compared to Proposition 4.3.3, the normality requirement on the reduction pair is dropped. This might look a minor improvement, but interestingly we have indeed constructed reduction pairs that are possibly not normal by the GWPO itself (Theorem 4.1.3).

We have a small remark on decidability. Assume that the signature is finite. Given reduction triples with each relation decidable, the induced relations $\geqslant_{\mathsf{G}'}$ and $>_{\mathsf{G}'}$ are also decidable. For example, checking if case 2c holds is done by checking $f(\bar{x}) \sqsupseteq_B t$ for each f with some fresh variables $\bar{x}$ not occurring in t, owing to the assumption that the underlying relations are stable. The same observation can be used when we solve the search problem of the GWPO satisfying certain ordering constraints by means of SMT/SAT solvers. For example, suppose that the encoding of the relation $\sqsupseteq_B$ is known for the designated class of reduction triples B. Then, the constraint that $f(\bar{x}) \sqsupseteq_B t$ for all function symbols f and variables $\bar{x}$ is equivalent to the finite conjunction of $f(\bar{z}) \sqsupseteq_B t$ for all function symbols f, where $\bar{z}$ are fresh variables not occurring in t.

From now on, we present a proof of Theorem 4.3.6. The proof follows the same line as Theorem 4.1.3, so we omit details but emphasize differences if there are. Let $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ be the GWPO induced by a partial status π and pairs $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ of relations on terms.

Lemma 4.3.8. *The inclusion $>_{\mathsf{G}'} \subseteq \geqslant_{\mathsf{G}'}$ holds.* ∎

Lemma 4.3.9. *Let $\rightsquigarrow$ be a stable relation on terms. Then $\rightsquigarrow = \mathcal{T} \times \mathcal{T}$ if and only if $x \rightsquigarrow y$ for some variables $x \neq y$.* ∎

Lemma 4.3.10. *If $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are stable order pairs of with $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$, then $\geqslant_{\mathsf{G}'}$ and $>_{\mathsf{G}'}$ are transitive, and $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ is compatible.*

Proof. We prove four properties simultaneously by induction on $|s| + |t| + |u|$: for all terms s, t and u if $s \geqslant_{\mathsf{G}'} t$ and $t \geqslant_{\mathsf{G}'} u$ then $s \geqslant_{\mathsf{G}'} u$; if $s \geqslant_{\mathsf{G}'} t$ and $t >_{\mathsf{G}'} u$ then $s >_{\mathsf{G}'} u$; if $s >_{\mathsf{G}'} t$ and $t \geqslant_{\mathsf{G}'} u$ then $s >_{\mathsf{G}'} u$; and if $s >_{\mathsf{G}'} t$ and $t >_{\mathsf{G}'} u$ then $s >_{\mathsf{G}'} u$. Here we only prove the first one since others can be proved in a similar way. Let $s \geqslant_{\mathsf{G}'} t$ and $t \geqslant_{\mathsf{G}'} u$. We distinguish how these are derived as in Table 4.1. In all cases we have $s \sqsupseteq_A u$ or $s \sqsupset_A u$ which we may not mention again.

A. Suppose that $s \geqslant_{\mathsf{G}'} t$ or $t \geqslant_{\mathsf{G}'} u$ follows from case 1. Then we have $s \sqsupset_A u$, so $s \geqslant_{\mathsf{G}'} u$ follows from case 1.

B. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2a. Then $s = f(s_1, \ldots, s_l)$ and $s_i \geqslant_{\mathsf{G}'} t$ for some $i \in \pi(f)$. By the induction hypothesis we have $s_i \geqslant_{\mathsf{G}'} u$, so $s \geqslant_{\mathsf{G}'} u$ follows from case 2a.

C. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2b and $t \geqslant_{\mathsf{G}'} u$ follows from 2a. Then, $t = g(t_1, \ldots, t_m)$ and there is some $j \in \pi(g)$ such that $s >_{\mathsf{G}'} t_j \geqslant_{\mathsf{G}'} u$. By the induction hypothesis and Lemma 4.3.8 we have $s \geqslant_{\mathsf{G}'} u$.

D. Suppose that both $s \geqslant_{\mathsf{G}'} t$ and $t \geqslant_{\mathsf{G}'} u$ follow from case 2b but at least one of them follows from case 2b(i). Then, $s \sqsupset_B u = h(u_1, \ldots, u_m)$ and $s \geqslant_{\mathsf{G}'} t >_{\mathsf{G}'} u_k$ for all $k \in \pi(h)$. So by the induction hypothesis, we have $s \geqslant_{\mathsf{G}'} u$ by case 2b(i).

E. Suppose that both $s \geqslant_{\mathsf{G}'} t$ and $t \geqslant_{\mathsf{G}'} u$ follow from case 2b(ii). Then

$$s = f(s_1, \ldots, s_l) \sqsupseteq_B t = g(t_1, \ldots, t_m) \sqsupseteq_B u = h(u_1, \ldots, u_n)$$

and $s \geqslant_{\mathsf{G}'} t >_{\mathsf{G}'} u_k$ for all $k \in \pi(h)$. Moreover,

$$\pi(f)(s_1, \ldots, s_l) \geqslant^{\mathsf{lex}}_{\mathsf{G}'} \pi(g)(t_1, \ldots, t_m) \geqslant^{\mathsf{lex}}_{\mathsf{G}'} \pi(h)(u_1, \ldots, u_n).$$

So we have $s \geqslant_{\mathsf{G}'} u$ by case 2b(ii) with the induction hypothesis.

F. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2b and $t \geqslant u$ from 2d. Since $s \sqsupset_B t$ or $s \sqsupseteq_B t$ holds, it is not hard to see that $s \geqslant_{\mathsf{G}'} u$ follows from case 2d.

G. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2c and $t \geqslant_{\mathsf{G}'} u$ from 2b(i). If $s = t$ then $s \geqslant_{\mathsf{G}'} u$ is trivial. So let $t = g(\bar{t}) \sqsupset_B h(\bar{u}) = u$ and assume that for all $f \in \mathcal{F}$ and variables $\bar{x}$ it holds that $f(\bar{x}) \sqsupseteq_B g(\bar{t})$. By substituting h and $\bar{u}$ into f and $\bar{x}$ respectively, we obtain $u \sqsupseteq_B t \sqsupset_B u$, a contradiction.

H. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2c and $t \geqslant_{\mathsf{G}'} u$ from 2b(ii). Then, $s \in \mathcal{V}$, $t = g(\bar{t}) \sqsupseteq_B h(\bar{u}) = u$, $\pi(g) = [\,]$, and $\pi(g)(\bar{t}) \geqslant^{\mathsf{lex}}_{\mathsf{G}'} \pi(h)(\bar{u})$. So $\pi(h) = [\,]$ as well. Therefore $s \geqslant_{\mathsf{G}'} u$ by case 2c.

I. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2c and $t \geqslant_{\mathsf{G}'} u$ from 2d. Here we have $s \sqsupseteq_A t \sqsupseteq_A u$ and $s, u \in \mathcal{V}$. If $s \neq u$ then by Lemma 4.3.9 we have a contradiction to $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$. So $s = u$ and we are done by case 2c.

J. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2d and $t \geqslant_{\mathsf{G}'} u$ from 2c. Then $s = f(\bar{s}) \sqsupseteq_A t \sqsupseteq_A h(u_1, \ldots, u_m) = u$, the relation $\sqsupset_A$ is simple with respect to π, and $s \sqsupset_B h(\bar{x})$ or $s \sqsupseteq_B h(\bar{x})$ and $\pi(h) = [\,]$ for some variables $\bar{x}$ not occurring in s. In the former case when $s \sqsupset_B h(\bar{x})$, by substituting $u_1, \ldots, u_m$ into $\bar{x}$

Table 4.1: Case distinction for the proof of Lemma 4.3.10.

	1	2a	2b(i)	2b(ii)	2c	2d
1	A	A	A	A	A	A
2a	A	B	B	B	B	B
2b(i)	A	C	D	D	easy	F
2b(ii)	A	C	D	E	easy	F
2c	A	easy	G	H	easy	I
2d	A	easy	easy	easy	J	easy

we obtain $s \sqsupseteq_B u$ and $s >_{\mathsf{G}'} u_k$ for all $k \in \pi(h)$ by the simplicity of $\sqsupset_A$ with respect to π. So $s \geqslant_{\mathsf{G}'} u$ by case 2b(i). In the latter case when $s \sqsupseteq_B h(\bar{x})$ and $\pi(h) = [\,]$, by substituting $u_1, \ldots, u_m$ into $\bar{x}$, we obtain $s \geqslant_{\mathsf{G}'} u$ by case 2b(ii).

In the remaining cases, it is rather easy to obtain the claim or contradictions. ∎

It is known that $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$ is essential for transitivity, see [103].

Lemma 4.3.11. *If $\sqsupseteq_A$ and $\sqsupseteq_B$ are reflexive and $\sqsupseteq_A$ is simple with respect to π, then*

- *$t >_{\mathsf{G}'} t_i$ whenever $i \in \pi(f)$ and $t = f(t_1, \ldots, t_n)$, and*
- *$t \geqslant_{\mathsf{G}'} t$ for all t (or $\geqslant_{\mathsf{G}'}$ is reflexive).* ∎

Next we show stability via Lemma 4.3.11. For convenience, we introduce a notation to denote modified substitutions. For a substitution σ, variables $x_1, \ldots, x_n$ and terms $t_1, \ldots, t_n$, the substitution $\sigma[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$ is defined by:

$$\sigma[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n](z) = \begin{cases} t_i & \text{if } z = x_i \\ \sigma(z) & \text{otherwise} \end{cases}$$

Lemma 4.3.12. *If $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are stable order pairs, and $\sqsupseteq_A$ is simple with respect to π, then $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ is stable.*

Proof. It suffices to prove the following properties: for all terms s and t and substitutions σ, if $s \geqslant_{\mathsf{G}'} t$ then $s\sigma \geqslant_{\mathsf{G}'} t\sigma$, and if $s >_{\mathsf{G}'} t$ then $s\sigma >_{\mathsf{G}'} t\sigma$. We prove these simultaneously by induction on $|s| + |t|$. Here we only prove the first property since the other can be proved in a similar way. To this end, we analyze how $s \geqslant_{\mathsf{G}'} t$ is derived.

1. If $s \geqslant_{\mathsf{G}'} t$ follows from case 1, then $s\sigma \sqsupset_A t\sigma$, so $s\sigma \geqslant_{\mathsf{G}'} t\sigma$ from case 1.
2. If $s \geqslant_{\mathsf{G}'} t$ follows from case 2, we have $s\sigma \sqsupseteq_A t\sigma$.
   a. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2a. By the induction hypothesis, we obtain $s\sigma \geqslant_{\mathsf{G}'} t\sigma$ again by case 2a.

b. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2b. Whether it is case 2b(i) or 2b(ii), with the induction hypothesis, we obtain $s\sigma \geqslant_{\mathsf{G}'} t\sigma$ again by case 2b.

c. Suppose that $s \geqslant_{\mathsf{G}'} t$ follows from case 2c. Let $s \in \mathcal{V}$. If $s = t$ then $s\sigma \geqslant_{\mathsf{G}'} t\sigma$ follows from Lemma 4.3.11. Otherwise, $t = g(t_1, \ldots, t_n)$, $\pi(g) = [\,]$ and $f(\bar{x}) \sqsupseteq_B t$ for all $f \in \mathcal{F}$ and variables $\bar{x}$. We further analyze whether $s\sigma \in \mathcal{V}$ or not.

   - Assume $s\sigma \in \mathcal{V}$. We argue towards case 2c: For all functions $f^{(m)}$ and variables $x_1, \ldots, x_m$, we choose some variables $y_1, \ldots, y_m$ that do not occur in t. We have $f(y_1, \ldots, y_m) \sqsupseteq_B t$ by assumption, so by applying $\sigma[y_1 \mapsto x_1, \ldots, y_m \mapsto x_m]$ we obtain $f(x_1, \ldots, x_m) \sqsupseteq_B t\sigma$ as desired. The remaining conditions are easy to verify.
   - Assume $s\sigma = f(s_1, \ldots, s_m)$. We argue towards case 2b(ii): Choosing some variables $y_1, \ldots, y_m$ that do not occur in t, we apply $\sigma[y_1 \mapsto s_1, \ldots, y_m \mapsto s_m]$ to $f(y_1, \ldots, y_m) \sqsupseteq_B t$. Then we obtain $f(s_1, \ldots, s_m) \sqsupseteq_B t\sigma$. As $\pi(g) = [\,]$ case 2b(ii) applies.

d. Suppose that $s = f(s_1, \ldots, s_m)$, $t \in \mathcal{V}$, the relation $\sqsupset_A$ is simple with respect to π, and for all $g \in \mathcal{F}$ and variables $\bar{x}$, the relation $s \sqsupset_B g(\bar{x})$ or $s \sqsupseteq_B g(\bar{x})$ and $\pi(g) = [\,]$.

   - Suppose $t\sigma \in \mathcal{V}$. We argue towards case 2d: Then, for all $f^{(n)} \in \mathcal{F}$ and $x_1, \ldots, x_n$, we choose some variables $y_1, \ldots, y_n$ that do not occur in s. By assumption, we have $s \sqsupset_B g(y_1, \ldots, y_n)$ or $s \sqsupseteq_B g(y_1, \ldots, y_n)$ and $\pi(g) = [\,]$. Applying $\sigma[y_1 \mapsto x_1, \ldots, y_n \mapsto x_n]$ we obtain $s\sigma \sqsupset_B g(x_1, \ldots, x_n)$ or $s\sigma \sqsupseteq_B g(x_1, \ldots, x_n)$ and $\pi(g) = [\,]$ as desired.
   - Suppose $t\sigma = g(t_1, \ldots, t_n)$. In this case we argue towards case 2b. We choose some variables $y_1, \ldots, y_n$ that do not occur in s. By assumption, we have $s \sqsupset_B g(y_1, \ldots, y_n)$ or $s \sqsupseteq_B g(y_1, \ldots, y_n)$ and $\pi(g) = [\,]$. In the former case, by applying $\sigma[y_1 \mapsto t_i, \ldots, y_n \mapsto t_n]$ we have $s\sigma \sqsupset_B t\sigma$ in addition to $s\sigma \sqsupseteq_A t\sigma \sqsupset_A t_j$ for all $j \in \pi(g)$. Since A is compatible, this yields $s\sigma >_{\mathsf{G}'} t_j$ for all $j \in \pi(g)$ by case 1. So $s\sigma \geqslant_{\mathsf{G}'} t\sigma$ by case 2b(i). In the latter case when $s \sqsupseteq_B g(y_1, \ldots, y_n)$ and $\pi(g) = [\,]$, by applying $\sigma[y_1 \mapsto t_i, \ldots, y_n \mapsto t_n]$ we have $s\sigma \sqsupseteq_B t\sigma$ so $s\sigma \geqslant_{\mathsf{G}'} t\sigma$ by case 2b(ii). ∎

Note that in the last case the compatibility of A is used. The following example show that this is essential.

Example 4.3.13. *Consider the signature* $\{\mathsf{a}^{(0)}, \mathsf{b}^{(0)}, \mathsf{f}^{(1)}\}$. *Let* $(\sqsupseteq_B, \sqsupset_B)$ *be the stable order pair induced by the precedence* $\mathsf{a} \equiv \mathsf{b} \succ \mathsf{f}$, *and let* π *be the partial status defined by* $\pi(\mathsf{f}) = [1]$ *and* $\pi(\mathsf{a}) = \pi(\mathsf{b}) = [\,]$. *We define* $\sqsupseteq_A$ *as* $\mathcal{T} \times \mathcal{T}$, *and* $s \sqsupset_A t$ *as* $s = \mathsf{f}(t)$. *The relations* $\sqsupseteq_A$ *and* $\sqsupset_A$ *are reflexive, transitive, stable, and simple with respect to* π *but not compatible, as* $\mathsf{f}(\mathsf{a}) \sqsupset_A \mathsf{a} \sqsupseteq_A \mathsf{b}$ *but* $\mathsf{f}(\mathsf{a}) \sqsupset_A \mathsf{b}$ *does not hold. We can observe*

that $\geqslant_{\mathsf{G}'}$ induced by π, A and B is not stable: $\mathsf{a} \geqslant_{\mathsf{G}'} x$ *by case 2d, but instantiating x with* $\mathsf{f}(\mathsf{b})$ *leads to* $\mathsf{a} \geqslant_{\mathsf{G}'} \mathsf{f}(\mathsf{b})$ *which we do not have.*

We can prove well-foundedness via Buchholz's method using Lemma 4.3.10.

Lemma 4.3.14. *Suppose that the signature is bounded, $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are well-founded stable order pairs, $\sqsupseteq_A$ is simple with respect to π, and $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$. Then $>_{\mathsf{G}'}$ is well-founded.* ∎

Monotonicity is easy to prove with Lemma 4.3.11. Let $(\geqslant_{\mathsf{MG}'}, >_{\mathsf{MG}'})$ be the monotone generalized weighted path order induced by a partial status π and triples $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ of relations on terms.

Lemma 4.3.15. *Suppose that $\geqslant_A$ and $\geqslant_B$ are monotone, $\sqsupseteq_A$ and $\sqsupseteq_B$ are reflexive and harmonious to $\geqslant_A$ and $\geqslant_B$, respectively, and $\sqsupseteq_A$ is simple with respect to π. Then $\geqslant_{\mathsf{MG}'}$ is monotone. If in addition π is total, then $>_{\mathsf{MG}'}$ is also monotone.* ∎

Theorem 4.3.6 is now obtained by invoking these lemmas. Moreover, the properties (Propositions 4.1.14 to 4.1.16) required to use the usable rule technique and lexicographic combination carry over to this refined version.

4.4 SPO as Reduction Pair — Refined Version

Theorem 4.3.6 is restricted to the case when $(\sqsupseteq_A, \sqsupset_A)$ is non-trivial. This becomes a problem if we want to obtain the SPO version by plugging the trivial reduction triple as in Section 4.2. This section is devoted to a way to avoid it.

We start with the refined version of the SPO corresponding to Definition 4.3.5 with A the trivial reduction triple.

Definition 4.4.1. *Let π be a partial status and let $(\sqsupseteq, \sqsupset)$ be a pair of relations on terms. The* semantic path order *$(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$ induced by π and $(\sqsupseteq, \sqsupset)$ is defined inductively as follows: $s \geqslant_{\mathsf{S}'} t$ if*

1. *$s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{S}'} t$ for some $i \in \pi(f)$.*
2. *$s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\mathsf{S}'} t_j$ for all $j \in \pi(g)$, and*
   a. *$s \sqsupset t$ or*
   b. *$s \sqsupseteq t$ and $\pi(f)(s_1, \ldots, s_m) \geqslant^{\mathsf{lex}}_{\mathsf{S}'} \pi(g)(t_1, \ldots, t_n)$.*
3. *$s \in \mathcal{V}$ and either $s = t$ or $t = g(\bar{t})$, $\pi(g) = [\,]$ and $f(\bar{x}) \sqsupseteq t$ for all $f \in \mathcal{F}$ and variables $\bar{x}$.*
4. *$s = f(\bar{s})$, $t \in \mathcal{V}$, π is empty, and for all function symbols g and variables $\bar{x}$, it holds that $s \sqsupset g(\bar{x})$ or $s \sqsupseteq g(\bar{x})$.*

The relation $>_{\mathsf{S}'}$ is defined by cases 1, 2a and 2b where $\geqslant^{\mathsf{lex}}_{\mathsf{S}'}$ is replaced by $>^{\mathsf{lex}}_{\mathsf{S}'}$. Here $(\geqslant^{\mathsf{lex}}_{\mathsf{S}'}, >^{\mathsf{lex}}_{\mathsf{S}'})$ is the lexicographic extension of $(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$. On top of these, for a partial status π and a reduction triple $(\geqslant, \sqsupseteq, \sqsupset)$ we define $\geqslant_{\mathsf{MS}'}$ as $\geqslant_{\mathsf{S}'} \cap \geqslant$ and $>_{\mathsf{MS}'}$ as $>_{\mathsf{S}'} \cap \geqslant$.

If the empty relation $\varnothing$ is simple with respect to π, the status π must be empty. So case 4 above precisely corresponds to case 2d in Definition 4.3.4.

Now, let us recall that $\geqslant_{\mathsf{MS}'}$ may not be transitive, as spotted in [103]:

Example 4.4.2. *Consider the signature $\{\mathsf{a}^{(0)}\}$ and let $(\geqslant, \sqsupseteq, \sqsupset)$ be the trivial reduction triple. We have $x \geqslant_{\mathsf{MS}'} \mathsf{a}$ by case 3 and also $\mathsf{a} \geqslant_{\mathsf{MS}'} y$ by case 4, but $x \geqslant_{\mathsf{MS}'} y$ does not hold.*

Actually, even in this case, $(\geqslant_{\mathsf{MS}'}, >_{\mathsf{S}'})$ forms a pseudo-reduction pair which can be used with dependency pairs (Theorem 2.7.3). The main obstacle to prove it is how to show that $(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$ is compatible. The idea is to avoid the pathological interaction of case 3 and case 4 as in Example 4.4.2. Let $(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$ be the SPO induced by a partial status π and a stable order pair $(\sqsupseteq, \sqsupset)$. We define $(\geqslant_{\mathsf{S}''}, >_{\mathsf{S}''})$ as follows: $s \geqslant_{\mathsf{S}''} t$ if $s \geqslant_{\mathsf{S}'} t$ can be derived without using case 4, and similarly, $s >_{\mathsf{S}''} t$ if $s >_{\mathsf{S}'} t$ can be derived without using case 4.

Lemma 4.4.3. *The pair $(\geqslant_{\mathsf{S}''}, >_{\mathsf{S}''})$ is an order pair.*

Proof. This can be proven by replaying the proof of Lemma 4.1.8. ∎

Observe that case 4 in Definition 4.4.1 is applicable only if π is empty, i.e., $\pi(f) = [\,]$ for all function symbols f. So, in the case when π is non-empty, $(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$ is identical to $(\geqslant_{\mathsf{S}''}, >_{\mathsf{S}''})$ by definition, which enables Lemma 4.4.3. The other case is handled by the following lemma.

Lemma 4.4.4. *If π is empty, then $(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$ is compatible.*

Proof. To show $>_{\mathsf{S}'} \cdot \geqslant_{\mathsf{S}'} \subseteq >_{\mathsf{S}'}$, let $s >_{\mathsf{S}'} t \geqslant_{\mathsf{S}'} u$. Since π is empty, $s >_{\mathsf{S}'} t$ is equivalent to $s = f(\bar{s}) \sqsupset g(\bar{t}) = t$. We analyze how $t \geqslant_{\mathsf{S}'} u$ is derived:

1. If $t \geqslant_{\mathsf{S}'} u$ follows from case 1, then it contradicts to the emptiness of π.
2. If $t \geqslant_{\mathsf{S}'} u$ follows from case 2, then $s >_{\mathsf{S}'} u$ follows from case 2a.
3. If $t \geqslant_{\mathsf{S}'} u$ follows from case 3, then it contradicts to $t = g(\bar{t})$.
4. If $t \geqslant_{\mathsf{S}'} u$ follows from case 4, for some fresh variables $\bar{x}$ we have $g(\bar{t}) \sqsupset f(\bar{x})$ or $g(\bar{t}) \sqsupseteq f(\bar{x})$. So by stability, we obtain $g(\bar{t}) \sqsupset f(\bar{s})$ or $g(\bar{t}) \sqsupseteq f(\bar{s})$, which leads to a contradiction with $f(\bar{s}) \sqsupset g(\bar{t})$.

To show $\geqslant_{\mathsf{S}'} \cdot >_{\mathsf{S}'} \subseteq >_{\mathsf{S}'}$, let $s \geqslant_{\mathsf{S}'} t >_{\mathsf{S}'} u$. Again, since π is empty, $t >_{\mathsf{S}'} u$ is equivalent to $t = g(\bar{t}) \sqsupset h(\bar{u}) = u$. We analyze how $s \geqslant_{\mathsf{S}'} t$ is derived:

1. If $s \geqslant_{\mathsf{S}'} t$ follows from case 1, then it contradicts to the emptiness of π.
2. If $s \geqslant_{\mathsf{S}'} t$ follows from case 2, then $s >_{\mathsf{S}'} u$ follows from case 2a.
3. If $s \geqslant_{\mathsf{S}'} t$ follows from case 3, then for some fresh variables $\bar{x}$ we have $h(\bar{x}) \sqsupseteq g(\bar{t})$, and by stability also $h(\bar{u}) \sqsupseteq g(\bar{t})$. This is a contradiction to $g(\bar{t}) \sqsupset h(\bar{u})$.
4. If $s \geqslant_{\mathsf{S}'} t$ follows from case 4, then it contradicts to $t = g(\bar{t})$. ∎

Putting these together, we have shown the compatibility of $(\geqslant_{\mathsf{S}'}, >_{\mathsf{S}'})$. Also, the other properties can be proven as in Section 4.3. In particular, well-foundedness is proven using only compatibility, see the proof of Lemma 4.1.11. So, we obtain the following result.

Theorem 4.4.5. *Suppose that the signature is bounded. Let π be a partial status and $(\geqslant, \sqsupseteq, \sqsupset)$ be a reduction triple. Then $(\geqslant_{\mathsf{MS}'}, \geqslant_{\mathsf{S}'})$ is a pseudo-reduction pair; if π is non-empty, then $(\geqslant_{\mathsf{MS}'}, \geqslant_{\mathsf{S}'})$ is a reduction pair; and if in addition π is total, then $(\geqslant_{\mathsf{MS}'}, \geqslant_{\mathsf{MS}'})$ is a monotone reduction pair.* ∎

What we have considered so far amounts to the missing case of Theorem 4.3.6, namely when $(\sqsupseteq_A, \sqsupset_A)$ is trivial. So, as a consequence, we obtain an improved version of it.

Corollary 4.4.6. *Suppose that the signature is bounded. Let π be a partial status, and $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ reduction triples such that $\sqsupseteq_A$ is simple with respect to π. Then $(\geqslant_{\mathsf{MG}'}, >_{\mathsf{G}'})$ is a pseudo-reduction pair; if $(\sqsupseteq_A, \sqsupset_A)$ is non-trivial or π is non-empty, then $(\geqslant_{\mathsf{MG}'}, >_{\mathsf{G}'})$ is a reduction pair; and if in addition π is total, then $(\geqslant_{\mathsf{MG}'}, >_{\mathsf{MG}'})$ is a monotone reduction pair.* ∎

From this, one can see that the non-triviality requirement of Theorem 4.3.2 can be dropped if pseudo-reduction pairs are enough.

We remark on the effect of having the refinements in terms of the simulation result (presented in Section 3.2). Actually, with these refinements, the SPO and the GWPO become incomparable, in the following sense: Suppose that $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ are order pairs with $\sqsupseteq_A$ simple with respect to a partial status π. Let $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ be the GWPO induced by π, A and B, and $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ the SPO induced by π and the lexicographic combination $(\sqsupseteq_{AB}, \sqsupset_{AB})$. The same example as Example 4.2.8 is enough to see that $\geqslant_{\mathsf{G}'} \subseteq \geqslant_{\mathsf{S}'}$ and $>_{\mathsf{G}'} \subseteq >_{\mathsf{S}'}$ may not hold. Even worse, the analog of Lemma 4.2.5 fails.

Example 4.4.7. *Let $\{\mathsf{f}^{(1)}\}$ be the signature, A the reduction triple $(\geqslant_{\mathcal{A}}, \geqslant_{\mathcal{A}}, >_{\mathcal{A}})$ induced by the algebra $\mathcal{A}$ on $\mathbb{N}$ with $\mathsf{f}_{\mathcal{A}}(x) = 1$, B the reduction triple induced by the precedence, and $\pi(\mathsf{f}) = []$. Now, we have $f(x) \sqsupseteq_{AB} \mathsf{f}(y)$ for all $x, y \in \mathcal{V}$. So $x \geqslant_{\mathsf{S}'} \mathsf{f}(y)$ by case 3. On the other hand, $x \geqslant_{\mathsf{G}'} \mathsf{f}(y)$ does not hold as neither $x >_{\mathcal{A}} \mathsf{f}(y)$ nor $x \geqslant_{\mathcal{A}} \mathsf{f}(y)$ holds.*

4.5 Evaluation with Benchmark Problems

In order to evaluate the GWPO and SPO as reduction pairs, we implemented a prototype termination tool based on the dependency pair method (Section 2.7). The experimental design is detailed in what follows, while basically it parallels that of Section 3.5 unless otherwise stated.

Construction of reduction pairs. The tool implements the following reduction pairs.

- `W`. WPO (Theorem 2.7.8) induced by a partial status π, a weakly π-simple $\mathcal{F}$-algebra $\mathcal{A}$ and a precedence
- `G`. GWPO $(\gtrsim_{\mathsf{MG}}, >_{\mathsf{G}})$ (Theorem 4.1.3) induced by a partial status π and reduction triples $(\gtrsim_{\mathcal{A}}, \gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\gtrsim_{\mathcal{B}}, \gtrsim^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$, where $\mathcal{A}$ is a weakly π-simple $\mathcal{F}$-algebra and $\mathcal{B}$ is an $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebra
- `S`. MSPO $(\gtrsim_{\mathsf{MS}}, >_{\mathsf{S}})$ (Corollary 4.2.3) induced by a partial status and the lexicographic combination of $(\gtrsim_{\mathcal{A}}, \gtrsim^{\sharp}_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}})$ and $(\gtrsim_{\mathcal{B}}, \gtrsim^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$ for $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebras $\mathcal{A}$ and $\mathcal{B}$

Here $\mathcal{F}$ is the signature of a given TRS and its dependency pairs, and $\mathcal{A}$ and $\mathcal{B}$ are linear polynomial interpretations over $\mathbb{N}$ or max/plus interpretations. In theory, for the same type of $\mathcal{A}$ and $\mathcal{B}$, if a TRS is proven terminating by `W`, then it is also shown by `G`. As we have seen in Section 4.2, `S` is incomparable with `W` and `G`.

Implementation. Apart from the aforementioned reduction pairs, our prototype termination tool is based on the standard techniques in the dependency pair method: recursive cycle analysis based on strongly connected components of dependency graphs [3, 45, 54], and the usable rule technique (with tcap [46] but without the refinement called usable rule w.r.t. argument filter [48]). As discussed at the end of Section 4.1, $\mathcal{C}_{\mathcal{E}}$-compatibility is guaranteed for free.

How to read the result table. Let us explain how to read the result table (Table 4.2) with an example: method `G` whose $\mathcal{A}$ is a max/plus interpretation and $\mathcal{B}$ is a linear polynomial interpretation over proved termination of 605 TRSs in the problem set, and reached time-limit for 90 problems; and the total runtime on the whole problem set was 155 minutes.

Table 4.2: Experiments on 1528 TRSs from TPDB 11.5.

		$\mathcal{B}$ is linear		$\mathcal{B}$ is max/plus
$\mathcal{A}$ is linear	W:	*486* (66 min, 29 TO)	W:	-
	G:	*514* (90 min, 45 TO)	G:	*591* (185 min, 104 TO)
	S:	*514* (105 min, 51 TO)	S:	*595* (197 min, 107 TO)
$\mathcal{A}$ is max/plus	W:	-	W:	*540* (134 min, 76 TO)
	G:	*605* (155 min, 90 TO)	G:	*551* (216 min, 123 TO)
	S:	*595* (211 min, 123 TO)	S:	*547* (278 min, 177 TO)

Improvement over the original WPO. Overall, one obtains more termination proofs by switching from the original WPO (W) to our methods. In particular, combining different kinds of algebras is quite effective. This is not possible for W. We note that not all systems proven terminating by W are shown by G, due to time-limit. Although G and S solved the same number (514) of problems when linear polynomials are used, they cover different sets of problems.

Comparison to state-of-the-art termination provers. Compared to state-of-the-art termination provers, every problem solved by our tool is also solved by AProVE or NaTT. So the author does not know if any new problem can be solved by our methods.

Incorporating rule removal. It is natural to ask what happens if the rule removal method [101] is used in the implementation. This is possible by using $(\succcurlyeq_{\mathsf{MG}}, >_{\mathsf{MG}})$ and $(\succcurlyeq_{\mathsf{MS}}, >_{\mathsf{MS}})$ instead of $(\succcurlyeq_{\mathsf{MG}}, >_{\mathsf{G}})$ and $(\succcurlyeq_{\mathsf{MS}}, >_{\mathsf{S}})$, respectively, in the construction of reduction pairs by Theorem 4.1.3 and Corollary 4.2.3. Unfortunately, we confirmed that, even so, each method solved zero or one more problems in the problem set.

Using refined versions. We have not implemented the refined versions of the GWPO and the SPO so far. However, we anticipate that the improvement is negligible in practice. For reference, in our setting, the refined WPO (already implemented in the prototype tool) solves fewer problems due to time-limit, regardless of linear polynomial or max/plus.

4.6 Related Work

We conclude by discussing related work.

Incorporating multiset extension. The path orders considered in this section compare arguments only lexicographically. In fact, the WPO formalized in Isabelle/HOL [95] extends Definition 4.3.1 so that it allows a choice between multiset and lexicographic extensions for each symbol, incorporating the well-known extension for the recursive path order. The author anticipates that the same extension is also possible for the refined GWPO (Definition 4.3.4), because the proofs presented in Sections 4.2 and 4.3 seem independent of lexicographic extension.

Unifying the SPO and the interpretation method. Example 4.2.8 also reveals that the SPO as reduction pair seems to hardly simulate the polynomial interpretation, in contrast to the (G)WPO. Similarly, it is obscure if the completeness result (Theorem 4.3.7) for non-trivial reduction pairs holds for the SPO. Interestingly, Borralleras and Rubio (who are originators of the MSPO) and their collaborators have investigated a different unification of the polynomial interpretation and the recursive path order [21] which splits the signature into two groups: the symbols in one are treated as in the polynomial interpretation, and those in the other are treated as in the recursive path order. Although this idea is quite different from that of the (G)WPO, it can be understood as a stepping stone to extending the SPO in an alternative way.

Chapter 5

Generalized Weighted Path Order as Co-rewrite Pair

In this chapter, we discuss co-rewrite pair, a generalization of reduction pair for non-reachability analysis of rewriting. It is known that the WPO can be regarded as a device to construct co-rewrite pairs. As expected, the GWPO can be used for the same purpose, and moreover it advances the capability of automated non-reachability analysis.

5.1 Generalized Weighted Path Order as Rewrite Pair

There is a rewrite pair version of Theorem 2.7.8 for the WPO which is obtained by dropping the well-foundedness assumptions.

Theorem 5.1.1 ([114, Proposition 1])**.** *Let π be a partial status, $\succeq$ a precedence, and $\mathcal{A}$ an algebra that is normal, weakly π-simple and weakly monotone. Then $(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$ is a rewrite pair.*

We illustrate a non-reachability proof by Proposition 2.8.3 with the theorem above.

Example 5.1.2. *Let $\mathcal{R}$ be the following TRS:*

$$\begin{aligned}
\mathsf{nth}(0, \mathsf{cons}(y, ys)) &\xrightarrow{1} y \qquad \mathsf{nth}(\mathsf{s}(x), \mathsf{cons}(y, ys)) \xrightarrow{2} \mathsf{nth}(x, ys) \\
\mathsf{nths}(\mathsf{nil}, ys) &\xrightarrow{3} \mathsf{nil} \qquad \mathsf{nths}(\mathsf{cons}(x, xs), ys) \xrightarrow{4} \mathsf{cons}(\mathsf{nth}(x, ys), \mathsf{nths}(xs, ys)) \\
&\qquad\qquad\qquad\qquad \mathsf{from}(x) \xrightarrow{5} \mathsf{cons}(x, \mathsf{from}(\mathsf{s}(x)))
\end{aligned}$$

Here $\mathsf{nth}(n, xs)$ extracts the n-th element from a list xs, so for example

$$\mathsf{nths}(\mathsf{cons}(0, \mathsf{cons}(\mathsf{s}^3(0), \mathsf{nil})), \mathsf{from}(\mathsf{s}(0))) \to_{\mathcal{R}}^* \mathsf{cons}(\mathsf{s}(0), \mathsf{cons}(\mathsf{s}^4(0), \mathsf{nil})).$$

We show that $x \twoheadrightarrow \mathsf{cons}(y, x)$ is $\mathcal{R}$-unsatisfiable. Let $\mathcal{A}$ be the algebra on $\mathbb{Z}_{\leqslant 0}$ with

$$\begin{aligned}
\mathsf{s}_{\mathcal{A}}(x) &= x - 1 & \mathsf{nil}_{\mathcal{A}} &= 0_{\mathcal{A}} = -1 & \mathsf{nth}_{\mathcal{A}}(x, y) &= 0 \\
\mathsf{cons}_{\mathcal{A}}(x, y) &= 0 & \mathsf{from}_{\mathcal{A}}(x) &= 0 & \mathsf{nths}_{\mathcal{A}}(x, y) &= 0
\end{aligned}$$

and π the partial status with $\pi(\mathsf{s}) = [\,]$, $\pi(\mathsf{nth}) = [2]$ and $\pi(f) = [1,\ldots,n]$ for other function symbols $f^{(n)}$. Since $\mathcal{A}$ is weakly simple with respect to π, we know that $(\geqslant_{\mathsf{W}}, >_{\mathsf{W}})$ induced by π, $\mathcal{A}$ and a precedence $\succeq$ with $\mathsf{nths} \succ \mathsf{nth} \succ \mathsf{from} \succ \mathsf{cons}$ is a rewrite pair. Now, we have $\mathcal{R} \subseteq \geqslant_{\mathsf{W}}$. Interesting to verify is rule 5: we first apply case 2b(i) with $\mathsf{from} \succ \mathsf{cons}$; then the obligations to show are $\mathsf{from}(x) >_{\mathsf{W}} x$ and $\mathsf{from}(x) >_{\mathsf{W}} \mathsf{from}(\mathsf{s}(x))$; the former is done with case 2a; for the latter we apply case 2b(ii), which demands to show $x >_{\mathsf{W}} \mathsf{s}(x)$; this is shown by case 1 as $\mathsf{s}_{\mathcal{A}}(x) = x - 1$. Moreover, for the reachability atom $x \twoheadrightarrow \mathsf{cons}(y, x)$ we also have $\mathsf{cons}(y, x) >_{\mathsf{W}} x$ as $2 \in \pi(\mathsf{cons})$. So by Proposition 2.8.3, we conclude that the reachability atom is unsatisfiable.

As we expect, there is also a rewrite pair version of Theorem 4.1.3 for the GWPO. Recall that the basic ingredients for the GWPO are reduction triples. In the rewrite pair setting, *rewrite triples* play the central role: a triple $(\geqslant, \sqsupseteq, \sqsupset)$ of relations on terms is a rewrite triple if $(\sqsupseteq, \sqsupset)$ is a stable order pair, $\geqslant$ is a rewrite preorder and $\sqsupseteq$ is harmonious to $\geqslant$. In other words, rewrite triple is reduction triple without well-foundedness. Consequently, every reduction triple is a rewrite triple.

By using rewrite triples for Theorem 4.1.3 instead of reduction triples, we obtain a rewrite pair instead of a reduction pair.

Theorem 5.1.3. *Let π be a partial status, and $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ rewrite triples. If $\sqsupseteq_A$ is simple with respect to π, then $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ is a rewrite pair.*

Theorem 5.1.1 is a special case obtained by instantiating A with an algebra and B with a precedence.

To prove Theorem 5.1.3, we need an additional lemma for irreflexivity. To this end, we follow the proof of [114, Lemma 3].

For a partial status π, we define $\trianglerighteq^{\pi}$ as the least reflexive and transitive relation on terms such that $f(s_1,\ldots,s_n) \trianglerighteq^{\pi} t$ if $s_i \trianglerighteq^{\pi} t$ for some $i \in \pi(f)$. It is easy to see that $\trianglerighteq^{\pi}$ is a quasi-order on terms, and the strict part $\triangleright^{\pi}$ is the least transitive relation such that $f(s_1,\ldots,s_n) \triangleright^{\pi} t$ if $s_i \trianglerighteq^{\pi} t$ for some $i \in \pi(f)$. Obviously, $\trianglerighteq$ and $\triangleright$ extend $\trianglerighteq^{\pi}$ and $\triangleright^{\pi}$, respectively.

Let $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ be the GWPO induced by pairs $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ of relations on terms.

Lemma 5.1.4. *If $\sqsupseteq_A$ is reflexive, transitive and simple with respect to π, then it holds that $\trianglerighteq^{\pi} \subseteq \sqsupseteq_A$.* ∎

Lemma 5.1.5. *Suppose that $(\sqsupseteq_A, \sqsupset_A)$ is an order pair with $\sqsupseteq_A$ simple with respect to π. Then $s \trianglerighteq^{\pi} t >_{\mathsf{G}} u$ implies $s >_{\mathsf{G}} u$ for all terms s, t and u.*

Proof. We perform induction on $|s|$, analyzing $s \trianglerighteq^{\pi} t$. If $s = t$ then we are done. So let $s = f(s_1,\ldots,s_n)$ and $s_i \trianglerighteq^{\pi} t$ for some $i \in \pi(f)$. By the induction

hypothesis and Lemma 4.1.7 we obtain $s_i \geqslant_{\mathsf{G}} u$. We have $s_i \sqsupset_A u$ or $s_i \sqsupseteq_A u$. In the former case, $s \sqsupseteq_A s_i \sqsupset_A u$ leads to $s >_{\mathsf{G}} u$ by case 1. In the latter case, $s \sqsupseteq_A s_i \sqsupseteq_A u$ leads to $s >_{\mathsf{G}} u$ by case 2a. ∎

Lemma 5.1.6. *If $(\sqsupseteq_A, \sqsupset_A)$ is an order pair, $\sqsupset_B$ is irreflexive, and $\sqsupseteq_A$ is simple with respect to π, then $>_{\mathsf{G}}$ is irreflexive.*

We follow the neat proof by nested induction due to [114, Lemma 3].

Proof. We prove that $t >_{\mathsf{G}} t$ leads to contradiction for all terms t. We perform induction on $|t|$, analyzing the derivation of $t >_{\mathsf{G}} t$. If $t >_{\mathsf{G}} t$ follows from case 1 or 2b(i), we obtain a contradiction by assumption. If $t >_{\mathsf{G}} t$ follows from case 2b(ii), we obtain a contradiction by the induction hypothesis. For the remaining case 2a, let $t = g(t_1, \ldots, t_m)$. For refuting this case, it suffices to prove a more general claim: $t \rhd^{\pi} s \geqslant_{\mathsf{G}} t$ is contradictory for all terms s. We prove this sublemma by induction on $|s|$, analyzing the derivation of $s \geqslant_{\mathsf{G}} t$.

1. Suppose $s \sqsupset_A t$. Then by Lemma 5.1.4 we have $t \sqsupseteq_A s \sqsupset_A t$, a contradiction.
2. Suppose $s \sqsupseteq_A t$.
   a. If $s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{G}} t$ for some $i \in \pi(f)$, then $t \rhd^{\pi} s_i \geqslant_{\mathsf{G}} t$, from which we obtain a contradiction with the inner induction hypothesis.
   b. If $s = f(s_1, \ldots, s_m)$ and $s >_{\mathsf{G}} t_j$ for all $j \in \pi(g)$, then there is some $j \in \pi(g)$ such that $t_j \trianglerighteq^{\pi} s >_{\mathsf{G}} t_j$. By Lemma 5.1.5 we obtain $t_j >_{\mathsf{G}} t_j$, from which we obtain a contradiction with the outer induction hypothesis.
   c. If $s = t$ we obtain a contradiction from $t \rhd^{\pi} s$. ∎

The lemma above and those proven in Section 4.1 are sufficient for proving Theorem 5.1.3. We obtain a rewrite pair theorem for the MSPO as a corollary of Theorem 5.1.3.

Corollary 5.1.7. *Let π be a partial status, and $(\geqslant, \sqsupseteq, \sqsupset)$ a rewrite triple. Then $(\geqslant_{\mathsf{MS}}, >_{\mathsf{S}})$ is a rewrite pair.*

Proof. The pair $(\geqslant_{\mathsf{MS}}, >_{\mathsf{S}})$ is identical to $(\geqslant_{\mathsf{MG}}, >_{\mathsf{G}})$ induced from the reduction triple $(\mathcal{T} \times \mathcal{T}, \mathcal{T} \times \mathcal{T}, \varnothing)$ and $(\geqslant, \sqsupseteq, \sqsupset)$. So Theorem 5.1.3 applies. ∎

The simulation result in Section 4.2 carries over to this setting, because it does not rely on well-foundedness. We also remark that the SPO cannot simulate the GWPO by any construction, see Example 4.2.10.

There are rewrite pair versions of Proposition 2.5.14 and Proposition 2.5.15.

Proposition 5.1.8. *Let $\mathcal{F}$ be the signature. For a weakly monotone $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebra $\mathcal{A}$, the triple $(\geqslant_{\mathcal{A}}, \geqslant^{\sharp}_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}})$ is a rewrite triple on $\mathcal{T}(\mathcal{F}, \mathcal{V})$.* ∎

Proposition 5.1.9. *Let $(\succcurlyeq_A, \sqsupseteq_A, \sqsupset_A)$ and $(\succcurlyeq_B, \sqsupseteq_B, \sqsupset_B)$ be rewrite triples. Then $(\succcurlyeq_A \cap \succcurlyeq_B, \sqsupseteq_{AB}, \sqsupset_{AB})$ is also a rewrite triple.* ∎

We illustrate Corollary 5.1.7 and Proposition 5.1.9 with an example. As discussed in Chapter 4, the SPO (Corollary 5.1.7) allows us to bypass the weak simplicity requirement.

Example 5.1.10. *Now, we show that $x \twoheadrightarrow \mathsf{s}(x)$ is unsatisfiable with respect to $\mathcal{R}$ of Example 5.1.2. Let $\mathcal{A}$ be the algebra on $\mathbb{Z}_{\leqslant 0}$ whose interpretations are given by*

$$\mathsf{s}_{\mathcal{A}}^{(\sharp)}(x) = x - 1 \quad \mathsf{nil}_{\mathcal{A}}^{(\sharp)} = 0_{\mathcal{A}}^{(\sharp)} = 0 \quad \mathsf{nth}_{\mathcal{A}}(x,y) = 0 \quad \mathsf{nth}_{\mathcal{A}}^{\sharp}(x,y) = -1$$
$$\mathsf{cons}_{\mathcal{A}}^{(\sharp)}(x,y) = y - 1 \quad \mathsf{from}_{\mathcal{A}}^{(\sharp)}(x) = x \quad \mathsf{nths}_{\mathcal{A}}(x,y) = x \quad \mathsf{nths}_{\mathcal{A}}^{\sharp}(x,y) = 0$$

and π the partial status with $\pi(\mathsf{nth}) = [2]$ and $\pi(f) = [1,\dots,n]$ for other function symbols $f^{(n)}$. Consider the rewrite pair $(\succcurlyeq_{\mathsf{MS}}, >_{\mathsf{S}})$ induced by π and the rewrite triple $(\succcurlyeq_{\mathcal{A}}, \succcurlyeq_{\mathcal{A}}^{\sharp}, >_{\mathcal{A}}^{\sharp})$. Then we have $\mathcal{R} \subseteq \succcurlyeq_{\mathsf{MS}}$. Interesting are rules 4 and 5. To see $\mathsf{nths}(\mathsf{cons}(x,xs),ys) \succcurlyeq_{\mathsf{MS}} \mathsf{cons}(\mathsf{nth}(x,ys),\mathsf{nths}(xs,ys))$ for rule 4, verifying

$$\overbrace{\mathsf{nths}(\mathsf{cons}(x,xs),ys)}^{xs-1} \succcurlyeq_{\mathcal{A}} \overbrace{\mathsf{cons}(\mathsf{nth}(x,ys),\mathsf{nths}(xs,ys))}^{xs-1}$$
$$\overbrace{\mathsf{nths}^{\sharp}(\mathsf{cons}(x,xs),ys)}^{0} >_{\mathcal{A}} \overbrace{\mathsf{cons}^{\sharp}(\mathsf{nth}(x,ys),\mathsf{nths}(xs,ys))}^{xs-1}, \quad \overbrace{\mathsf{nth}^{\sharp}(x,ys)}^{-1}$$

we finally compare $\mathsf{nths}(\mathsf{cons}(x,xs),ys) >_{\mathsf{S}} \mathsf{nths}(xs,ys)$, which is done by case 2b. Rule 5 is handled easily with $\mathsf{cons}_{\mathcal{A}}^{(\sharp)}(x,y) = y - 1$ and $\mathsf{s}_{\mathcal{A}}^{(\sharp)}(x) = x - 1$. Moreover, for the atom $x \twoheadrightarrow \mathsf{s}(x)$ we also have $\mathsf{s}(x) >_{\mathsf{S}} x$ as $1 \in \pi(\mathsf{s})$. So by Proposition 2.8.3, we conclude that the reachability atom is unsatisfiable. This proof does not go through if we use the WPO instead, because the first argument of $\mathsf{s}_{\mathcal{A}}(x) = x - 1$ is not weakly simple while $1 \in \pi(\mathsf{s})$.

We remark that any of the automatic tools from the 2024 and 2025 editions of the Confluence Competition (which has a dedicated category for a variant of the reachability problem) [51, 52] could not prove the unsatisfiability of Example 5.1.10.

Finally, we conclude a rewrite pair version of Corollary 4.4.6.

Corollary 5.1.11. *Let π be a partial status, and $(\succcurlyeq_A, \sqsupseteq_A, \sqsupset_A)$ and $(\succcurlyeq_B, \sqsupseteq_B, \sqsupset_B)$ rewrite triples with $\sqsupseteq_A$ simple with respect to π. If $(\sqsupseteq_A, \sqsupset_A)$ is non-trivial or π is non-empty, then $(\succcurlyeq_{\mathsf{MG}'}, >_{\mathsf{G}'})$ is a rewrite pair.*

Proof. We have proved everything except for irreflexivity of $>_{\mathsf{G}'}$, which can be proven in the same way as Lemma 5.1.6. ∎

So, we could use the refined version for constructing rewrite pairs.

5.2 Generalizing Co-weighted Path Order

It is known that the WPO as rewrite pair is further refined to the *co-weighted path order* so as to enjoy the generality of co-rewrite pair [114]. The aim of this section is to demonstrate that this result is also obtained as a corollary of a more general theorem about GWPO.

Following Yamada [114], we first introduce a dual of order pairs induced by algebras. Let $\mathcal{A}$ be an algebra endowed with an order pair $(\geqslant, >)$. We write $s \geqslant_{\bar{\mathcal{A}}} t$ if $[\alpha](s) \not< [\alpha](t)$ for all assignments α, and similarly, $s >_{\bar{\mathcal{A}}} t$ if $[\alpha](s) \not\leqslant [\alpha](t)$ for all assignments α[1]. The notations might look odd, but we note that if $(\geqslant, >)$ is total and normal then $\geqslant$ equals $> \cup =$ and therefore $(\not<, \not\leqslant)$ is identical to $(\geqslant, >)$. So, in this case $(\geqslant_{\bar{\mathcal{A}}}, >_{\bar{\mathcal{A}}})$ is identical to $(\geqslant_{\mathcal{A}}, >_{\mathcal{A}})$. An algebra on $\mathbb{N}$ ordered as usual are such an example. We note that these relations are different from the complements of $\geqslant_{\mathcal{A}}$ and $>_{\mathcal{A}}$. For example, neither $x \geqslant_{\mathcal{A}} y$ nor $x \geqslant_{\bar{\mathcal{A}}} y$ holds for an algebra $\mathcal{A}$ on $\mathbb{N}$.

Definition 5.2.1. *Let π be a partial status, $\mathcal{A}$ an algebra endowed with an order pair $(\geqslant, >)$, and $\succeq$ a precedence.*[2] *The* co-weighted path order *(co-WPO) $(\geqslant_{\overline{\mathsf{W}}}, >_{\overline{\mathsf{W}}})$ is the pair of relations on terms defined simultaneously as follows: $s \geqslant_{\overline{\mathsf{W}}} t$ if*

1. $s >_{\bar{\mathcal{A}}} t$, *or*

2. $s \geqslant_{\bar{\mathcal{A}}} t$ *and one of the following conditions holds:*
   a. $s = f(s_1, \ldots, s_m)$ *and* $s_i \geqslant_{\overline{\mathsf{W}}} t$ *for some* $i \in \pi(f)$.
   b. $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $s >_{\overline{\mathsf{W}}} t_j$ *for all* $j \in \pi(g)$, *and*
      i. $f \not\succeq g$ *or*
      ii. $f \not\prec g$ *and* $\pi(f)(s_1, \ldots, s_m) \geqslant^{\mathsf{lex}}_{\overline{\mathsf{W}}} \pi(g)(t_1, \ldots, t_n)$.
   c. $s \in \mathcal{V}$ *and* $s = t$

The relation $>_{\overline{\mathsf{W}}}$ is defined by cases 1, 2a and 2b where $\geqslant^{\mathsf{lex}}_{\overline{\mathsf{W}}}$ is replaced by $>^{\mathsf{lex}}_{\overline{\mathsf{W}}}$. Here $(\geqslant^{\mathsf{lex}}_{\overline{\mathsf{W}}}, >^{\mathsf{lex}}_{\overline{\mathsf{W}}})$ is the lexicographic extension of $(\geqslant_{\overline{\mathsf{W}}}, >_{\overline{\mathsf{W}}})$.

Theorem 5.2.2 ([114, Proposition 2])**.** *Let π be a partial status, $\succeq$ a precedence, and $\mathcal{A}$ an algebra that is normal, weakly π-simple and weakly monotone. Then $(\geqslant_{\mathsf{W}}, >_{\overline{\mathsf{W}}})$ is a co-rewrite pair.*

A nice example to facilitate understanding is [114, Example 7], which tells us that the greater relations need not be the more profitable as negation is involved.

[1] In the original presentation [114], these are written as $\mathcal{A} \vDash s \not< t$ and $\mathcal{A} \vDash s \not\leqslant t$, respectively.
[2] We could consider precedence as order pair [114].

Example 5.2.3. *Let* $\mathcal{R} = \{\mathsf{a} \to \mathsf{b} \Leftarrow \mathsf{b} \twoheadrightarrow \mathsf{a}\}$. *To prove that* $\mathsf{a} \twoheadrightarrow \mathsf{b}$ *is unsatisfiable, we use* $(\geqslant_{\mathsf{W}}, >_{\overline{\mathsf{W}}})$ *induced by the trivial algebra and the identity precedence* $\succeq$. *Since* a *and* b *are incomparable with respect to* $\succeq$, *we have* $\mathsf{a} \not\succeq \mathsf{b} \not\succeq \mathsf{a}$ *and thereby* $\mathsf{a} >_{\overline{\mathsf{W}}} \mathsf{b} >_{\overline{\mathsf{W}}} \mathsf{a}$. *So with Theorem 5.2.2 we conclude that* $\mathsf{a} \twoheadrightarrow \mathsf{b}$ *is unsatisfiable.*

The key behind this theorem is *co-compatibility*. Formally, pairs $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ of relations are co-compatible if both $\sqsupseteq_A \cap \sqsubset_B = \varnothing$ and $\sqsupseteq_B \cap \sqsubset_A = \varnothing$ hold. By definition, every order pair is co-compatible with itself ([114, Proposition 3]).

To prove a GWPO version of Theorem 5.2.2, we need a co-compatibility lemma analogous to [114, Lemma 6]. Let π be a partial status, and let $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ be normal order pairs such that $\sqsupseteq_A$ is simple with respect to a partial status π. In addition, let $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ be pairs of relations on terms. Then we consider $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ induced by π, A and B and $(\geqslant_{\overline{\mathsf{G}}}, >_{\overline{\mathsf{G}}})$ induced by π, $\bar{A}$ and $\bar{B}$.

Lemma 5.2.4. *If* $(\sqsupseteq_A, \sqsupset_A)$ *and* $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ *as well as* $(\sqsupseteq_B, \sqsupset_B)$ *and* $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ *are co-compatible, so are* $(\geqslant_{\mathsf{G}}, >_{\mathsf{G}})$ *and* $(\geqslant_{\overline{\mathsf{G}}}, >_{\overline{\mathsf{G}}})$.

Proof. We prove that for all terms s and t, neither $s \geqslant_{\mathsf{G}} t >_{\overline{\mathsf{G}}} s$ nor $t >_{\mathsf{G}} s \geqslant_{\overline{\mathsf{G}}} t$ holds. This is done by proving these two claims simultaneously by induction on $|s| + |t|$. For the first claim that $s \geqslant_{\mathsf{G}} t >_{\overline{\mathsf{G}}} s$ does not hold, we do case distinction as in Table 5.1.

I. If $s \geqslant_{\mathsf{G}} t$ or $t >_{\overline{\mathsf{G}}} s$ follows from case 1, then we obtain a contradiction from the co-compatibility of A and $\bar{A}$, using the normality of A if necessary.

II. If $t >_{\overline{\mathsf{G}}} s$ follows from case 2a, then $t = g(t_1, \ldots, t_n)$ and $t_j \geqslant_{\overline{\mathsf{G}}} s$ for some $j \in \pi(g)$. By Lemma 4.1.9 we have $t >_{\mathsf{G}} t_j$, and then by Lemma 4.1.8 we obtain $s >_{\mathsf{G}} t_j$. Now we apply the induction hypothesis to obtain a contradiction.

III. If $s \geqslant_{\mathsf{G}} t$ follows from case 2a and $t >_{\overline{\mathsf{G}}} s$ from case 2b, then $s = f(s_1, \ldots, s_m)$ and $s_i \geqslant_{\mathsf{G}} t >_{\overline{\mathsf{G}}} s_i$ for some $i \in \pi(f)$, which is contradictory by the induction hypothesis.

IV. If both $s \geqslant_{\mathsf{G}} t$ and $t >_{\overline{\mathsf{G}}} s$ follows from case 2b but one of them from case 2b(i), then we obtain a contradiction from the co-compatibility of B and $\bar{B}$, using the normality of B if necessary.

V. If both $s \geqslant_{\mathsf{G}} t$ and $t >_{\overline{\mathsf{G}}} s$ follows from case 2b(ii), we appeal to the induction hypothesis with the fact that lexicographic extension preserves co-compatibility [114, Lemma 5].

VI. The remaining cases demand $s = t \in \mathcal{V}$ and $t \notin \mathcal{V}$, which is impossible.

Table 5.1: Case distinction for the proof of Lemma 5.2.4.

	1	2a	2b(i)	2b(ii)
1	I	I	I	I
2a	I	II	III	III
2b(i)	I	II	IV	IV
2b(ii)	I	II	IV	V
2c	I	II	VI	VI

The other claim is shown in a similar fashion. ∎

For the lemma above it is assumed that A and B are both normal. The following example shows that this assumption is essential: Let the signature be $\{\mathsf{a}^{(0)}, \mathsf{b}^{(0)}\}$. We define $s \sqsupseteq_A t$ and $s \sqsupseteq_{\bar{A}} t$ as $s = t$. Furthermore, we define $s \sqsupset_A t$ as $s = \mathsf{a}$ and $t = \mathsf{b}$, and $s \sqsupset_{\bar{A}} t$ as $s = \mathsf{b}$ and $t = \mathsf{a}$. Let B and $\bar{B}$ be $(\mathcal{T} \times \mathcal{T}, \varnothing)$. These order pairs satisfy all the assumptions except for the normality of A. The co-compatibility does not hold as $\mathsf{a} \geqslant_{\mathsf{G}} \mathsf{b} >_{\overline{\mathsf{G}}} \mathsf{a}$ by case 1. Similarly, by swapping the orders (i.e., using B, $\bar{B}$, A and $\bar{A}$ as A, $\bar{A}$, B and $\bar{B}$) we can show that the normality of B is essential.

Now, we introduce the notion of *co-rewrite quintuple* which underpins our co-rewrite pair theorem: A quintuple $(\geqslant_A, \sqsupseteq_A, \sqsupset_A, \sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ is a co-rewrite quintuple if $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ is a rewrite triple, $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ is stable, $\sqsupseteq_{\bar{A}}$ is reflexive, $\sqsupset_A \subseteq \sqsupseteq_A$ and finally, $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ are co-compatible. The co-rewrite quintuple is *simple with respect to* a partial status π if $\sqsupseteq_A$ and $\sqsupseteq_{\bar{A}}$ are simple with respect to π.

Theorem 5.2.5. *Let $(\geqslant_A, \sqsupseteq_A, \sqsupset_A, \sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B, \sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ be co-rewrite quintuples such that the quintuple A is simple w.r.t. a partial status π. Then $(\geqslant_{\mathsf{MG}}, >_{\overline{\mathsf{G}}})$ is a co-rewrite pair, where $\geqslant_{\mathsf{MG}}$ is induced by π, $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$, and $>_{\overline{\mathsf{G}}}$ is by π, $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$.*

Proof. That $\geqslant_{\mathsf{MG}}$ is a rewrite preorder follows from lemmas from Section 4.1. It follows that $\geqslant_{\mathsf{MG}} \cap <_{\overline{\mathsf{G}}} = \varnothing$ from Lemma 5.2.4. Finally, the stability of $>_{\overline{\mathsf{G}}}$ follows from Lemma 4.1.10. ∎

Like Corollary 5.1.11, we can incorporate the refined GWPO (Section 4.3) into the theorem above. A tricky thing here is that, we construct $\geqslant$ via the refined GWPO but $>$ without the refinements for constructing a co-rewrite pair $(\geqslant, >)$. The reason to do so is explained later.

Toward the goal, we prove a variant of Lemma 5.2.4: Let π be a partial status, and let $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_B, \sqsupset_B)$ be normal order pairs such that $\sqsupseteq_A$ is simple with respect to a partial status π. In addition, let $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ be

pairs of relations on terms. Then we consider $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ induced by π, A and B and $(\geqslant_{\overline{\mathsf{G}}}, >_{\overline{\mathsf{G}}})$ induced by π, $\bar{A}$ and $\bar{B}$.

Lemma 5.2.6. *If $(\sqsupseteq_A, \sqsupset_A)$ and $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ as well as $(\sqsupseteq_B, \sqsupset_B)$ and $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ are co-compatible, so are $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ and $(\geqslant_{\overline{\mathsf{G}}}, >_{\overline{\mathsf{G}}})$.*

Proof. Following the proof of Lemma 5.2.4, by induction on $|s| + |t|$ we prove that for all terms s and t, neither $s \geqslant_{\mathsf{G}'} t >_{\overline{\mathsf{G}}} s$ nor $t >_{\mathsf{G}'} s \geqslant_{\overline{\mathsf{G}}} t$ holds. In addition to the case analysis of Table 5.1, it suffices to consider the following cases.

- Suppose that $s \geqslant_{\mathsf{G}'} t$ by case 2c and $s \neq t$. So $s \in \mathcal{V}$, and $t = g(\bar{t})$ with $s \sqsupseteq_A t$ and $\pi(g) = [\,]$. If $t >_{\overline{\mathsf{G}}} s$ by case 1, we obtain a contradiction from the co-compatibility of A and $\bar{A}$; if $t >_{\overline{\mathsf{G}}} s$ by case 2a, we obtain a contradiction from $\pi(g) = [\,]$; and finally, if $t >_{\overline{\mathsf{G}}} s$ by case 2b, we obtain a contradiction from $s \in \mathcal{V}$.
- Suppose that $s \geqslant_{\mathsf{G}'} t$ by case 2d. So $s = f(\bar{s}) \sqsupseteq_A t \in \mathcal{V}$. Then $t >_{\overline{\mathsf{G}}} s$ must follow from case 1. In this case, we obtain a contradiction from the co-compatibility of A and $\bar{A}$.

We note that the case corresponding to case II of Table 5.1 can be handled by Lemma 4.3.11 and the compatibility of $(\geqslant_{\mathsf{G}'}, >_{\mathsf{G}'})$ (cf. Corollary 4.4.6). ∎

Theorem 5.2.7. *Let $(\geqslant_A, \sqsupseteq_A, \sqsupset_A, \sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B, \sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ be co-rewrite quintuples such that the quintuple A is simple w.r.t. a partial status π. If $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$ or π is non-empty, then $(\geqslant_{\mathsf{MG}'}, >_{\overline{\mathsf{G}}})$ is a co-rewrite pair. Here, $\geqslant_{\mathsf{MG}'}$ is induced by π, $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ and $(\geqslant_B, \sqsupseteq_B, \sqsupset_B)$ via Definition 4.3.4, and $>_{\overline{\mathsf{G}}}$ is by π, $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ via Definition 4.1.1.*

Proof. That $\geqslant_{\mathsf{MG}'}$ is a rewrite preorder can be seen from Corollary 5.1.11, and the stability of $>_{\overline{\mathsf{G}}}$ is shown by Lemma 4.1.10. Finally, $\geqslant_{\mathsf{MG}'} \cap <_{\overline{\mathsf{G}}} = \varnothing$ is shown by Lemma 5.2.6. ∎

Note that Lemma 4.3.12 (the stability lemma for the refined GWPO) demands order-pairedness in contrast to Lemma 4.1.10 (that for the GWPO without the refinements). Moreover, in the theorem, $(\sqsupseteq_{\bar{A}}, \sqsupset_{\bar{A}})$ and $(\sqsupseteq_{\bar{B}}, \sqsupset_{\bar{B}})$ need not be order pairs. This is why we use both versions of GWPO together.

The techniques to construct rewrite triples carriers over to co-rewrite quintuples. First of all, rewrite triples can be turned into co-rewrite quintuples.

Proposition 5.2.8. *The quintuple $(\geqslant_A, \sqsupseteq_A, \sqsupset_A, \sqsupseteq_A, \sqsupset_A)$ is a co-rewrite quintuple if $(\geqslant_A, \sqsupseteq_A, \sqsupset_A)$ is a rewrite triple with $\sqsupset_A \subseteq \sqsupseteq_A$.* ∎

So, the version of Theorem 5.1.3 with $\sqsupset_A \subseteq \sqsupseteq_A$ and $\sqsupset_B \subseteq \sqsupseteq_B$ is obtained by Theorem 5.2.5 and Proposition 5.2.8. Since Lemma 5.2.4 demands the normality assumptions, we have proven the theorems separately.

A trivial co-rewrite quintuple similar to Proposition 2.5.16 can be considered.

Proposition 5.2.9. *The quintuple* $(\mathcal{T}\times\mathcal{T},\mathcal{T}\times\mathcal{T},\varnothing,\mathcal{T}\times\mathcal{T},\varnothing)$ *is a co-rewrite quintuple.* ∎

As one can expect, we obtain a co-rewrite pair version of the SPO by using the trivial co-rewrite quintuple.

Corollary 5.2.10. *Let* $(\geqslant_A,\sqsupseteq_A,\sqsupset_A,\sqsupseteq_{\bar{A}},\sqsupset_{\bar{A}})$ *be a co-rewrite quintuple and* π *a partial status. Then* $(\geqslant_{\mathsf{MS}},>_{\bar{\mathsf{S}}})$ *is a co-rewrite pair. Here,* $(\geqslant_{\mathsf{MS}},>_{\mathsf{MS}})$ *is induced by* π *and* $(\geqslant_A,\sqsupseteq_A,\sqsupset_A)$*, and* $(\geqslant_{\bar{\mathsf{S}}},>_{\bar{\mathsf{S}}})$ *is by* π *and* $(\sqsupseteq_{\bar{A}},\sqsupset_{\bar{A}})$*.* ∎

We continue to transfer more results from the previous section. Not surprisingly, precedence is a simple but convenient way of constructing co-rewrite quintuples.

Proposition 5.2.11. *Let* $\succeq$ *be a precedence. Define* $(\sqsupseteq_P,\sqsupset_P)$ *as in Proposition 2.5.11,* $s\sqsupseteq_{\bar{P}}t$ *by* $s=t$ *or* $s=f(\bar{s}),t=g(\bar{t})$ *and* $f\not\prec g$*, and* $s\sqsupset_{\bar{P}}t$ *by* $s=f(\bar{s}),t=g(\bar{t})$ *and* $f\not\preceq g$*. Then* $(\mathcal{T}\times\mathcal{T},\sqsupseteq_P,\sqsupset_P,\sqsupseteq_{\bar{P}},\sqsupset_{\bar{P}})$ *is a co-rewrite quintuple.* ∎

It is instructive to examine Example 5.2.3 again with our techniques: To prove that $\mathsf{a}\twoheadrightarrow\mathsf{b}$ is unsatisfiable again, we use $(\geqslant_{\mathsf{MS}},>_{\bar{\mathsf{S}}})$ (Corollary 5.2.10) induced by the identity precedence $\succeq$ (Proposition 5.2.11). Since a and b are incomparable with respect to $\succeq$, we have $\mathsf{a}\sqsupset_{\bar{P}}\mathsf{b}\sqsupset_{\bar{P}}\mathsf{a}$ and thereby $\mathsf{a}>_{\bar{\mathsf{S}}}\mathsf{b}>_{\bar{\mathsf{S}}}\mathsf{a}$. So with Proposition 2.8.3 we conclude that $\mathsf{a}\twoheadrightarrow\mathsf{b}$ is unsatisfiable. Here, it is essential that a and b are incomparable with respect to $\succeq$. Putting it differently, although the SPO is incremental, the construction of Proposition 5.2.11 is not incremental since $\succeq$ is used in the negated forms.

The next is a construction based on algebra.

Proposition 5.2.12. *If* $\mathcal{A}$ *is a weakly monotone and normal algebra, the quintuple* $(\geqslant_{\mathcal{A}},\geqslant_{\mathcal{A}},>_{\mathcal{A}},\geqslant_{\bar{\mathcal{A}}},>_{\bar{\mathcal{A}}})$ *is a co-rewrite quintuple. Moreover, if* $\mathcal{A}$ *is weakly simple w.r.t. a partial status* π *then the co-rewrite quintuple is simple w.r.t.* π*.*

Proof. This is straightforward. The simplicity claim owes to [114, Lemma 5]. ∎

Now, the co-WPO theorem (Theorem 5.2.2) is a special case of Theorem 5.2.5 where A is constructed by Proposition 5.2.12 and B is by Proposition 5.2.11. Moreover, as Proposition 5.1.8 does, we can incorporate marking of root function symbols into Proposition 5.2.12: Let $\mathcal{F}$ be the signature, and $\mathcal{A}$ an $(\mathcal{F}\cup\mathcal{F}^\sharp)$-algebra. We define $s\geqslant^\sharp_{\bar{\mathcal{A}}}t$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ by $s=t$ or $s=f(\bar{s})$, $t=g(\bar{t})$ and $f^\sharp(\bar{s})\geqslant_{\bar{\mathcal{A}}}g^\sharp(\bar{t})$. Similarly, we define $s>^\sharp_{\bar{\mathcal{A}}}t$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ by $s=f(\bar{s})$, $t=g(\bar{t})$ and $f^\sharp(\bar{s})>_{\bar{\mathcal{A}}}g^\sharp(\bar{t})$.

Proposition 5.2.13. *The quintuple* $(\geqslant_{\mathcal{A}},\geqslant^\sharp_{\mathcal{A}},>^\sharp_{\mathcal{A}},\geqslant^\sharp_{\bar{\mathcal{A}}},>^\sharp_{\bar{\mathcal{A}}})$ *is a co-rewrite quintuple if* $\mathcal{A}$ *is normal and weakly monotone.* ∎

If $\mathcal{A}$ is normal and total, then $(\gtrsim^\sharp_{\overline{\mathcal{A}}}, >^\sharp_{\overline{\mathcal{A}}})$ degenerates to $(\gtrsim^\sharp_{\mathcal{A}}, >^\sharp_{\mathcal{A}})$. The same goes for Proposition 5.2.12.

Similarly, as Proposition 5.1.9 does, lexicographic combination is also useful.

Proposition 5.2.14. *Suppose that* $(\gtrsim_A, \sqsupseteq_A, \sqsupset_A, \sqsupseteq_{\overline{A}}, \sqsupset_{\overline{A}})$ *and* $(\gtrsim_B, \sqsupseteq_B, \sqsupset_B, \sqsupseteq_{\overline{B}}, \sqsupset_{\overline{B}})$ *are co-rewrite quintuples. Then* $(\gtrsim_A \cap \gtrsim_B, \sqsupseteq_{AB}, \sqsupset_{AB}, \sqsupseteq_{\overline{AB}}, \sqsupset_{\overline{AB}})$ *is also a co-rewrite quintuple.*

Proof. To prove that $\sqsupseteq_{AB} \cap \sqsubset_{\overline{AB}}$ is empty, let $s \sqsupseteq_{AB} t$ and $t \sqsupset_{\overline{AB}} s$. If $s \sqsupset_A t$ or $s \sqsupset_{\overline{A}} t$ holds we obtain a contradiction by co-compatibility and $\sqsupset_A \subseteq \sqsupseteq_A$. Otherwise, we use co-compatibility of B. The other claims are readily proven. ∎

Every co-rewrite pair can be turned into a co-rewrite quintuple.

Proposition 5.2.15. *If* $(\gtrsim, >)$ *is a co-rewrite pair, then* $(\gtrsim, \gtrsim, \varnothing, \mathcal{T} \times \mathcal{T}, >)$ *is a co-rewrite quintuple.* ∎

A co-rewrite pair $(\gtrsim, >)$ is *trivial* if $\gtrsim = \mathcal{T} \times \mathcal{T}$. Such a co-rewrite pair is useless for Proposition 2.8.3, because $> = \varnothing$ follows from $\gtrsim \cap < = \varnothing$. In fact, Proposition 5.2.15 allows us to simulate every non-trivial co-rewrite pair by a GWPO, with the help of Theorem 5.2.7. This is an analog of Theorem 4.3.7 for co-rewrite pair.

Theorem 5.2.16. *Let* $(\gtrsim, >)$ *be a non-trivial co-rewrite pair. Then there are a partial status* π *and co-rewrite quintuples* A *and* B *with* $\sqsupseteq_A \subsetneq \mathcal{T} \times \mathcal{T}$ *and* A *simple w.r.t.* π *such that* $(\gtrsim_{\mathsf{MG}'}, >_{\overline{\mathsf{G}}})$ *constructed as in Theorem 5.2.7 is identical to* $(\gtrsim, >)$.

Proof. Let π be the empty status, and let A be $(\gtrsim, \gtrsim, \varnothing, \mathcal{T} \times \mathcal{T}, >)$ and B the trivial co-rewrite quintuple. By an argument similar to that used for Theorem 4.3.7, we can show that $\gtrsim_{\mathsf{MG}'}$ is identical to $\gtrsim$. Moreover, $>_{\mathsf{G}}$ is induced by the empty status π, and the pairs $(\mathcal{T} \times \mathcal{T}, >)$ and $(\mathcal{T} \times \mathcal{T}, \varnothing)$. So, only case 1 is applicable and therefore $>_{\mathsf{G}}$ is identical to $>$. ∎

We note that the key construction (Proposition 5.2.15) in the proof utilizes the generality of co-rewrite quintuple, and the algebra-based formulation of the co-WPO is insufficient to accommodate it. In other words, Proposition 5.2.15 cannot be substituted by Proposition 5.2.12 which underlies the co-WPO, because it is restricted to a single algebra endowed with an order pair $(\gtrsim, >)$ and the negations $(\not<, \not\gtrsim)$.

5.3 Evaluation with Benchmark Problems

This section is devoted to experimental evaluation of our results with a prototype implementation. We first describe the design of the experiment, and then discuss the results.

Construction of co-rewrite pairs. The tool implements the following three kinds of co-rewrite pairs.

- `W`. WPO (Theorem 5.1.1) induced by an $\mathcal{F}$-algebra $\mathcal{A}$ and a precedence
- `G`. GWPO $(\gtrsim_{\mathsf{MG}}, >_{\mathsf{G}})$ (Theorem 5.1.3) induced by a partial status π and triples $(\gtrsim_{\mathcal{A}}, \gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$ and $(\gtrsim_{\mathcal{B}}, \gtrsim^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$, where $\mathcal{A}$ is a weakly π-simple $\mathcal{F}$-algebra and $\mathcal{B}$ is an $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebra
- `S`. MSPO $(\gtrsim_{\mathsf{MS}}, >_{\mathsf{S}})$ (Corollary 5.1.7) induced by a partial status and the lexicographic combination of $(\gtrsim_{\mathcal{A}}, \gtrsim^{\sharp}_{\mathcal{A}}, >^{\sharp}_{\mathcal{A}})$ and $(\gtrsim_{\mathcal{B}}, \gtrsim^{\sharp}_{\mathcal{B}}, >^{\sharp}_{\mathcal{B}})$ for $(\mathcal{F} \cup \mathcal{F}^{\sharp})$-algebras $\mathcal{A}$ and $\mathcal{B}$

Here $\mathcal{F}$ is the signature of a given TRS, and $\mathcal{A}$ and $\mathcal{B}$ are linear polynomial interpretations over $\mathbb{N}$ or $\mathbb{Z}_{\leqslant 0}$. In theory, for the same type of $\mathcal{A}$ and $\mathcal{B}$, if a reachability problem is proven unsatisfiable by the WPO, then it is also shown by the GWPO. In contrast, `S` is incomparable with `W` and `G` in theory. Although for `G` and `S` we could use the co-rewrite pair theorems (Theorem 5.2.5 and Corollary 5.2.10) instead of Theorem 5.1.3 and Corollary 5.1.7, in our simple setting it makes no difference as we only use total and normal algebras here.[3]

Implementation. Our prototype tool works as follows: Given a conditional TRS $\mathcal{R}$ and reachability atoms ϕ, we apply Proposition 2.8.2 to obtain a single reachability atom $s \twoheadrightarrow t$. Then, the tool searches a co-rewrite pair $(\gtrsim, >)$ that witnesses $\mathcal{R}$-unsatisfiability of $s \twoheadrightarrow t$ (Proposition 2.8.3) within the designated class (`W`, `G` or `S`) of co-rewrite pairs. As in Section 3.5, for linear polynomial interpretation (whether over $\mathbb{N}$ or $\mathbb{Z}_{\leqslant 0}$) the coefficient of each monomial is restricted to 0 or 1. On top of these, following the automation techniques of KBO [119] and WPO [117], the search is done by encoding the constraints into linear arithmetic expressions and then calling the SMT solver Z3 [31]. The experiment was run on a machine equipped with an Intel Core i7-11850H CPU (8 cores, 16 threads, 2.50–4.80 GHz) and 16 GB RAM.

Problem set. Our experiment used 146 reachability problems in the ARI-COPS database [2]. For reference, among those, the 2025 version of `infChecker` [50] (winning in the Confluence Competition since 2019) proved unsatisfiability of 64 problems, and satisfiability of 43. The 2024 version of `NaTT` (which implements various co-rewrite pairs [114]) proved unsatisfiability of 41 problems.

[3] It is future work to test non-total algebras such as the matrix interpretation [37].

Table 5.2: Experiments on 146 Reachability Problems from ARI-COPS.

		$\mathcal{B}$ on $\mathbb{N}$		$\mathcal{B}$ on $\mathbb{Z}_{\leqslant 0}$
$\mathcal{A}$ on $\mathbb{N}$	W:	*27* (6 min)	W:	-
	G:	*29* (6 min)	G:	*29* (5 min)
	S:	*29* (5 min)	S:	*29* (5 min)
$\mathcal{A}$ on $\mathbb{Z}_{\leqslant 0}$	W:	-	W:	*26* (6 min)
	G:	*30* (5 min)	G:	*31* (7 min)
	S:	*29* (5 min)	S:	*30* (6 min)

How to read the table. Table 5.2 indicates the numbers of problems proven unsatisfiable (with the total runtime) by the aforementioned methods. In this setting, our tool did not exceed the time-limit. So, as in theory, every problem solved by W was solved by the corresponding G.

Improvement over the WPO as rewrite pair. Table 5.2 shows that a handful of problems can be additionally solved by switching to the GWPO or SPO from the original WPO. In our setting, using G with interpretations on $\mathbb{Z}_{\leqslant 0}$ for both $\mathcal{A}$ and $\mathcal{B}$ solves the largest number of problems. This forms a contrast to the results in Sections 3.5 and 4.5, where using different kinds of interpretations for $\mathcal{A}$ and $\mathcal{B}$ works better.

Improvement over state-of-the-art tools. There are two problems that were solved by our prototype tool but not by any of infChecker and NaTT: Problem 1627[4] (Example 5.1.2) was solved by W, G and S, and Problem 1628 (Example 5.1.10) was by G and S, by using linear polynomial interpretation over $\mathbb{Z}_{\leqslant 0}$. We note that these problems are added to the database by the author.

Improvement over the co-WPO. Although G with linear polynomial interpretations over $\mathbb{N}$ solves 29 problems, precisely the same set of problems is solved if we instead use Theorem 5.2.2 (with precedence as order pair). On the other hand, if we use interpretations on $\mathbb{Z}_{\leqslant 0}$, the co-WPO solves 28 problems which are a proper subset of those solved by G.

5.4 Related Work

Discrimination pair due to Aoto is another generalization of rewrite pair which is useful to prove non-joinability (and thereby non-confluence of TRSs) [1]. Formally, a pair $(\geqslant, >)$ of relations on terms is a discrimination pair if $\geqslant$ is a rewrite

[4]The number is the problem ID number in ARI-COPS.

relation, $>$ is irreflexive and $\geqslant \cdot > \subseteq >$. So, since every rewrite pair is a discrimination pair by definition, the techniques developed in Section 5.1 are useful for non-joinability proving. An interesting line of future work is to refine our technique to fully enjoy the generality of discrimination pairs $(\geqslant, >)$, which does not demand stability of $>$.

So far we have compared our GWPO with other methods for reachability (or feasibility) analysis only using existing implementations as black boxes. More detailed comparison with general frameworks [50, 94] and techniques based on tree automata [38, 39, 77] and logical models [74] is left as future work.

Chapter 6

Conclusion

We conclude by adding about related and future work.

Completion procedures. Central in equational reasoning is the word problem which asks, given an equational theory $\mathcal{E}$ and an equation $s \approx t$, whether $s \leftrightarrow_{\mathcal{E}}^* t$ or not. The Knuth–Bendix completion [63] is a procedure to transform an equational theory $\mathcal{E}$ into a complete (terminating and confluent) TRS $\mathcal{R}$ with conversion equivalence $\leftrightarrow_{\mathcal{E}}^* = \leftrightarrow_{\mathcal{R}}^*$ (i.e., equivalence regarding the word problem). If the procedure is successful, the resulting TRS provides a decision procedure for the word problem of $\mathcal{E}$. A difficulty is that the completion procedure takes a reduction order as an additional parameter, which ensures the termination of $\mathcal{R}$ during the transformation. A challenging example is the theory of group with two commuting endomorphisms

$$\begin{aligned} (x+y)+z &= x+(y+z) & \mathsf{f}(x+y) &= \mathsf{f}(x)+\mathsf{f}(y) \\ (-x)+x &= 0 & \mathsf{g}(x+y) &= \mathsf{g}(x)+\mathsf{g}(y) \\ 0+x &= x & \mathsf{f}(x)+\mathsf{g}(y) &= \mathsf{g}(y)+\mathsf{f}(x) \end{aligned}$$

one of whose complete and conversion-equivalent TRSs is the twenty rules in Example 3.1.6 [96]. Running the Knuth–Bendix completion with a KBO or LPO is doomed to fail as the rule $\mathsf{f}(x)+\mathsf{g}(y) = \mathsf{g}(y)+\mathsf{f}(x)$ cannot be oriented. So this kind of challenging completion tasks have been done by variants of completion procedures which leverage automatic termination provers for TRSs instead of reduction orders [107, 112]. An alternative solution suggested by Hirokawa [53] is to use stronger reduction orders such as monotonic semantic path orders in combination with maximal completion [62]. The results developed in Chapter 3 should shine with this approach (see Example 3.1.6).

Ordered completion and superposition calculus. Ordered completion [10] is an extension of the Knuth–Bendix completion to a semi-decision procedure for the word problem, meaning that $s \leftrightarrow_{\mathcal{E}}^* t$ is eventually shown if it indeed holds.

Moreover, the method has been extended to superposition calculus [9], a semi-decision procedure for the validity problem in first-order logic. As the Knuth–Bendix completion does, both of the methods take a reduction order as an additional parameter. This parameter plays a crucial role in both theory and practice. On the theory side, the semi-decidability relies on ground-totality of the order. On the practical side, the order prunes the search space and therefore determines if the procedure succeeds in a reasonable amount of time. As discussed in Section 3.3, the generalized weighted path order can be used to construct an exotic yet ground-total reduction order which orients the combinator equation $((\mathsf{S} \cdot x) \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$ from *right to left*. This forms a sharp contrast to Bhayat and Reger's order [17] which orients it from *left to right*. As they remark, combinators are used for leveraging first-order theorem provers in proof assistants [19]. We therefore anticipate that our order also contributes to efficiency in such a setting.

Dependency pair method. The aim of Chapter 4 was to develop a generalization of the weighted path order and semantic path order as reduction pairs within the dependency pair method. On the other hand, Borralleras showed out that the monotonic semantic path order can simulate a basic version of the dependency pair method ([22, Section 4.4.2]), and also the dependency graph techniques ([22, Chapter 7]). It is worth investigating if our generalized weighted path order gives more insight.

Reduction pairs and reduction triples. Reduction pairs serve as decreasing measures for the dependency pair method, whereas reduction triples are better suited to the theory of monotonic semantic path orders and generalized weighted path orders. It is interesting to ask whether ideas from one setting can be carried over to the other. For example, we have already confirmed that the lexicographic combination technique [88] for reduction pairs is also useful to construct more sophisticated reduction triples. In fact, the monotonic semantic path order with this technique yields a reduction order which allows completion of another challenging completion task, namely the proof reduction system due to Wehrman and Stump ([106, Appendix A]).

Semantic labeling. Another popular method for proving termination is Zantema's semantic labeling [121]. Geser observed that the semantic path order corresponds to semantic labeling applied with the recursive path order [43]. Then, it is natural to ask what is the counter-part of the generalized weighted path order in semantic labeling. We anticipate that, for example, semantic labeling applied with the Knuth–Bendix order (as in [53]) can be simulated by the generalized weighted path order. Another interesting line of research is investigation

of the methods ([85, Section 5.5.2] and [99, Chapter 7]) to use semantic labeling within the dependency pair method. It is worth investigating the relationship between their methods and the generalized weighted path order (or semantic path order).

Completeness results. The refined version of the GWPO is capable of simulating every non-trivial reduction pair (Theorem 4.3.7), and also every non-trivial co-rewrite pair (Theorem 5.2.16). These results can be situated as analogs to the completeness of the MSPO (as reduction order) for termination: Given a terminating TRS $\mathcal{R}$, there is a reduction triple such that the induced MSPO $>_{\mathsf{MS}}$ satisfies $\mathcal{R} \subseteq >_{\mathsf{MS}}$ [26].

Formalization in proof assistants. The WPO has been formalized [103] in the proof assistant Isabelle/HOL [83], and thanks to this, one can verify the correctness of a termination proof with the WPO by means of the certifier CeTA which is extracted from the formalization library IsaFoR [100]. The WPO helps to keep a formalization library succinct yet versatile, since the single method covers many other techniques such as the KBO and the LPO. The same can be said for the GWPO, as we have seen in the thesis. So far, the author has formalized only a small part of the results about the MSPO (Theorem 2.5.10 and particular constructions of reduction triples). Formalizing the remaining results will require substantial engineering effort, because it inherently demands adjustments to interfaces shared among IsaFoR: Although there is already an interface for reduction triples (which is used for the MSPO formalization), adapting existing orders to reduction triples is yet to be done. If one aims to formalize the co-rewrite pairs (such as Theorem 5.2.7) too, the first task will be to design an interface for co-rewrite quintuples.

Extension to other rewriting formalisms. While we have only considered rewriting with terms in the most basic form (called first-order terms), there are advanced formalisms that are built on richer languages. Among others, higher-order rewriting considers extensions of the λ-calculus which allow variable binding in terms, see [97, Chapter 11] for an overview. We anticipate that our techniques can be incorporated into the MSPO [23] for higher-order rewriting. Similarly, as remarked in Section 3.6, we believe that further investigation of the semantic path order for rewriting modulo AC [24, 84] is a fruitful line of future work.

Ordering constraint solving. The ordering constraint solving problem [27, 29, 65, 82] asks to find a substitution satisfying a quantifier-free logical formula about a fixed reduction order. It is known that (ground) confluence of ordered

rewriting (under a signature extension) is decidable [30] under the assumption that the constraint solving problem of the underlying (ground-total) reduction order is decidable. Also, ordering constraint solving is used [110] for solving the ground joinability problem [76]. So, investigation of the constraint solving problem for the GWPO (equivalently, for the SPO) is apparently of interest.

From termination to complexity. It is known that from certain forms of termination proofs one can extract information about upper bounds of *derivational complexity*, i.e., the maximum number of steps needed to normalize terms of a certain size. On one hand, the LPO induces multiply recursive upper bounds [28, 109], the multiset path order induces primitive recursive ones [28, 58], and the KBO induces upper bounds expressed by the Ackermann function [70, 81]. On the other hand, the SPO (resp. WPO) characterizes (resp. simple) termination of TRSs, and therefore we cannot extract any meaningful information about derivational complexity, see [71, 108]. So, a possible question is whether we can restrict the (G)WPO in a way that it induces revealing complexity upper bounds. Moreover, there are variants of path orders [5–7] that are useful to (automatically) analyze a restricted form of derivational complexity (innermost *runtime* complexity). It seems also interesting to consider incorporating the idea of GWPO to those techniques.

References

[1] T. Aoto. Disproving confluence of term rewriting systems by interpretation and ordering. In *Proc. 9th International Symposium on Frontiers of Combining Systems*, volume 8152 of *LNCS*, pages 311–326, 2013. doi: 10.1007/978-3-642-40885-4_22.

[2] ARI Project. ARI-COPS Database. https://ari-cops.uibk.ac.at/ARI/. Accessed: 05.11.2025.

[3] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000. doi: 10.1016/S0304-3975(99)00207-8.

[4] T. Arts and J. Giesl. A collection of examples for termination of term rewriting using dependency pairs. Technical report, RWTH Aachen, 2001.

[5] M. Avanzini and G. Moser. Polynomial path orders. *Logical Methods in Computer Science*, 9(4), 2013. doi: 10.2168/LMCS-9(4:9)2013.

[6] M. Avanzini, N. Eguchi, and G. Moser. A path order for rewrite systems that compute exponential time functions. In *Proc. 22nd International Conference on Rewriting Techniques and Applications*, volume 10 of *LIPIcs*, pages 123–138, 2011. doi: 10.4230/LIPICS.RTA.2011.123.

[7] M. Avanzini, N. Eguchi, and G. Moser. A new order-theoretic characterisation of the polytime computable functions. *Theoretical Computer Science*, 585:3–24, 2015. doi: 10.1016/J.TCS.2015.03.003.

[8] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. doi: 10.1017/CBO9781139172752.

[9] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3): 217–247, 1994. doi: 10.1093/LOGCOM/4.3.217.

[10] L. Bachmair, N. Dershowitz, and D. A. Plaisted. *Resolution of Equations in Algebraic Structures: Completion without Failure*, volume 2, pages 1–30. Academic Press, 1989. doi: 10.1016/B978-0-12-046371-8.50007-9.

[11] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Elsevier, 1984.

[12] C. Barrett, P. Fontaine, and C. Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). https://smt-lib.org/, 2016.

[13] H. Becker, J.C. Blanchette, U. Waldmann, and D. Wand. A transfinite Knuth–Bendix order for lambda-free higher-order terms. In *Proc. 26th International Conference on Automated Deduction*, volume 10395 of *LNCS*, pages 432–453, 2017. doi: 10.1007/978-3-319-63046-5_27.

[14] A. Bentkamp. The embedding path order for lambda-free higher-order terms. *Journal of Applied Logics*, 8(10):2447–2470, 2021. URL https://collegepublications.co.uk/ifcolog/?00052.

[15] A. Bentkamp, J.C. Blanchette, V. Nummelin, S. Tourret, and U. Waldmann. Complete and efficient higher-order reasoning via lambda-superposition. *ACM SIGLOG News*, 10(4):25–40, 2023. doi: 10.1145/3636362.3636367.

[16] J. Bergstra and J.W. Klop. Church–Rosser strategies in the lambda calculus. *Theoretical Computer Science*, 9(1):27–38, 1979. doi: 10.6092/ISSN.1972-5787/4593.

[17] A. Bhayat and G. Reger. A Knuth–Bendix-like ordering for orienting combinator equations. In *Proc. 10th International Joint Conference on Automated Reasoning*, volume 12166 of *LNCS*, pages 259–277, 2020. doi: 10.1007/978-3-030-51074-9_15.

[18] A. Bhayat and G. Reger. A combinator-based superposition calculus for higher-order logic. In *Proc. 10th International Joint Conference on Automated Reasoning*, volume 12166 of *LNCS*, pages 278–296, 2020. doi: 10.1007/978-3-030-51074-9_16.

[19] J.C. Blanchette, C. Kaliszyk, L.C. Paulson, and J. Urban. Hammering towards QED. *Journal of Formalized Reasoning*, 9(1):101–148, 2016. doi: 10.6092/ISSN.1972-5787/4593.

[20] J.C. Blanchette, U. Waldmann, and D. Wand. A lambda-free higher-order recursive path order. In *Proc. 20th International Conference on Foundations of Software Science and Computation Structures*, volume 10203 of *LNCS*, pages 461–479, 2017. doi: 10.1007/978-3-662-54458-7_27.

[21] M. Bofill, C. Borralleras, E. Rodríguez-Carbonell, and A. Rubio. The recursive path and polynomial ordering for first-order and higher-order terms. *Journal of Logic and Computation*, 23(1):263–305, 2013. doi: 10.1093/LOGCOM/EXS027.

[22] C. Borralleras. *Ordering-Based Methods for Proving Termination Automatically*. PhD thesis, Universitat Politècnica de Catalunya, 2003.

[23] C. Borralleras and A. Rubio. A monotonic higher-order semantic path ordering. In *Proc. 8th International Conference on Logic Programming and Automated Reasoning*, volume 2250 of *LNCS*, pages 531–547, 2001. doi: 10.1007/3-540-45653-8_37.

[24] C. Borralleras and A. Rubio. Monotonic AC-compatible semantic path orderings. In *Proc. 14th International Conference on Rewriting Techniques and Applications*, volume 2706 of *LNCS*, pages 279–295, 2003. doi: 10.1007/3-540-44881-0_20.

[25] C. Borralleras and A. Rubio. Orderings and constraints: Theory and practice of proving termination. In *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, volume 4600 of *LNCS*, pages 28–43, 2007. doi: 10.1007/978-3-540-73147-4_2.

[26] C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. In *Proc. 17th International Conference on Automated Deduction*, volume 1831 of *LNCS (LNAI)*, pages 346–364, 2000. doi: 10.1007/10721959_27.

[27] Y. Briefs, H. Leidinger, and C. Weidenbach. KBO constraint solving revisited. In *Proc. 14th International Symposium on Frontiers of Combining Systems*, volume 14279 of *LNCS*, pages 81–98, 2023. doi: 10.1007/978-3-031-43369-6_5.

[28] W. Buchholz. Proof-theoretic analysis of termination proofs. *Annals of Pure and Applied Logic*, 75(1–2):57–65, 1995. doi: 10.1016/0168-0072(94)00056-9.

[29] H. Comon. Solving symbolic ordering constraints. *International Journal of Foundations of Computer Science*, 1(4):387–412, 1990. doi: 10.1142/S0129054190000278.

[30] H. Comon, P. Narendran, R. Nieuwenhuis, and M. Rusinowitch. Deciding the confluence of ordered term rewrite systems. *ACM Transactions on Computational Logic*, 4(1):33–55, 2003. doi: 10.1145/601775.601777.

[31] L. M. de Moura and N. S. Bjørner. Z3: an efficient SMT solver. In *Proc. 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *LNCS*, pages 337–340, 2008. doi: 10.1007/978-3-540-78800-3_24.

[32] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17:279–301, 1982. doi: 10.1016/0304-3975(82)90026-3.

[33] N. Dershowitz. 33 Examples of termination. In *TCS School 1993: Term Rewriting*, volume 909 of *LNCS*, pages 16–26, 1995. doi: 10.1007/3-540-59340-3_2.

[34] N. Dershowitz. Termination by abstraction. In *Proc. 20th International Conference on Logic Programming*, volume 3132 of *LNCS*, pages 1–18, 2004. doi: 10.1007/978-3-540-27775-0_1.

[35] N. Dershowitz and C. Hoot. Natural termination. *Theoretical Computer Science*, 142:179–207, 1995. doi: 10.1016/0304-3975(94)00275-4.

[36] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979. doi: 10.1145/359138.359142.

[37] J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. *Journal of Automated Reasoning*, 40 (2-3):195–220, 2008. doi: 10.1007/S10817-007-9087-9.

[38] B. Felgenhauer and R. Thiemann. Reachability, confluence, and termination analysis with state-compatible automata. *Information and Computation*, 253:467–483, 2017. doi: 10.1016/J.IC.2016.06.011.

[39] G. Feuillade, T. Genet, and V.V.T. Tong. Reachability analysis over term rewriting systems. *Journal of Automated Reasoning*, 33(3-4):341–383, 2004. doi: 10.1007/S10817-004-6246-0.

[40] C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. SAT solving for termination analysis with polynomial interpretations. In *Proc. 10th International Conference on Theory and Applications in Satisfiability Testing*, volume 4501 of *LNCS*, pages 340–354, 2007. doi: 10.1007/978-3-540-72788-0_33.

[41] C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. Maximal termination. In *Proc. 19th International Conference on Rewriting Techniques and Applications*, volume 5117 of *LNCS*, pages 110–125, 2008. doi: 10.1007/978-3-540-70590-1_8.

[42] A. Geser. On a monotonic semantic path ordering. Technical Report 92–13, Ulmer Informatik-Berichte, Universität Ulm, Germany, 1992.

[43] A. Geser. Semantic labelling followed by recursive path order with status is strictly more powerful than the corresponding semantic path order. Technical report, Universität Passau, 1994.

[44] A. Geser. An improved general path order. *Applicable Algebra in Engineering, Communication and Computing*, 7:469–511, 1996. doi: 10.1007/BF01293264.

[45] J. Giesl, T. Arts, and E. Ohlebusch. Modular termination proofs for rewriting using dependency pairs. *Journal of Symbolic Computation*, 34:21–58, 2002. doi: 10.1006/jsco.2002.0541.

[46] J. Giesl, R. Thiemann, and P. Schneider-Kamp. Proving and disproving termination of higher-order functions. In *Proc. 5th International Symposium on Frontiers of Combining Systems*, volume 3717 of *LNCS (LNAI)*, pages 216–231, 2005. doi: 10.1007/11559306_12.

[47] J. Giesl, R. Thiemann, and P. Schneider-Kamp. The dependency pair framework: Combining techniques for automated termination proofs. In *Proc. 11th International Conference on Logic Programming and Automated Reasoning*, volume 3452 of *LNCS (LNAI)*, pages 301–331, 2005. doi: 10.1007/978-3-540-32275-7_21.

[48] J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37:155–203, 2006. doi: 10.1007/s10817-006-9057-7.

[49] J. Giesl, C. Aschermann, M. Brockschmidt, F. Emmes, F. Frohn, C. Fuhs, J. Hensel, C. Otto, M. Plücker, P. Schneider-Kamp, T. Ströder, S. Swiderski, and R. Thiemann. Analyzing program termination and complexity automatically with AProVE. *Journal of Automated Reasoning*, 58:3–31, 2017. doi: 10.1007/s10817-016-9388-y.

[50] R. Gutiérrez and S. Lucas. Automatically proving and disproving feasibility conditions. In *Proc. 10th International Joint Conference on Automated Reasoning*, volume 12167 of *LNCS*, pages 416–435, 2020. doi: 10.1007/978-3-030-51054-1_27.

[51] R. Gutiérrez, A. Middeldorp, N. Nishida, T. Saito, and R. Thiemann. Confluence competition 2024. In *Proc. 13th Workshop on Confluence*, page 53, 2024. The proceedings are available at https://iwc24.github.io/IWC2024_proceedings.pdf.

[52] R. Gutiérrez, A. Middeldorp, N. Nishida, T. Saito, and R. Thiemann. Confluence competition 2025. In *Proc. 14th Workshop on Confluence*,

page 52, 2025. The proceedings are available at https://iwc24.github.io/IWC2024_proceedings.pdf.

[53] N. Hirokawa. Completion and reduction orders. In *Proc. 6th International Conference on Formal Structures on Computation and Deduction*, volume 195 of *LIPIcs*, pages 2:1–2:9, 2021. doi: 10.4230/LIPIcs.FSCD.2021.2.

[54] N. Hirokawa and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199(1-2):172–199, 2005. doi: 10.1016/J.IC.2004.10.004.

[55] N. Hirokawa and A. Middeldorp. Tyrolean termination tool: Techniques and features. *Information and Computation*, 205(4):474–511, 2007. doi: 10.1016/J.IC.2006.08.010.

[56] N. Hirokawa and A. Middeldorp. Hydra battles and AC termination, revisited. In *Proc. 19th Workshop on Termination*, pages 1–6, 2023. The paper is available at https://arxiv.org/abs/2307.14036v1.

[57] N. Hirokawa and A. Middeldorp. Hydra battles and AC termination. *Logical Methods in Computer Science*, 21(2), 2025. doi: 10.46298/LMCS-21(2:29)2025.

[58] D. Hofbauer. Termination proofs by multiset path orderings imply primitive recursive derivation lengths. *Theoretical Computer Science*, 105(1):129–140, 1992. doi: 10.1016/0304-3975(92)90289-R.

[59] S. Kamin and J.J. Lévy. Two generalizations of the recursive path ordering. Technical report, University of Illinois, 1980. Unpublished manuscript.

[60] J.-C. Kassing, 2025. Personal communication.

[61] D. Kim, T. Saito, R. Thiemann, and A. Yamada. An Isabelle formalization of co-rewrite pairs for non-reachability in term rewriting. In *Proc. 14th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 272–282, 2025. doi: 10.1145/3703595.3705889.

[62] D. Klein and N. Hirokawa. Maximal completion. In *Proc. 22nd International Conference on Rewriting Techniques and Applications*, volume 10 of *LIPIcs*, pages 71–80, 2011. doi: 10.4230/LIPIcs.RTA.2011.71.

[63] D.E. Knuth and P.B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970. doi: 10.1016/B978-0-08-012975-4.50028-X.

[64] K. Korovin and A. Voronkov. An AC-compatible Knuth–Bendix order. In *Proc. 19th International Conference on Automated Deduction*, volume 2741 of *LNCS (LNAI)*, pages 45–59, 2003. doi: 10.1007/978-3-540-45085-6_5.

[65] K. Korovin and A. Voronkov. Knuth–Bendix constraint solving is NP-complete. *ACM Transactions on Computational Logic*, 6(2):361–388, 2005. doi: 10.1145/1055686.1055692.

[66] M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean Termination Tool 2. In *Proc. 20th International Conference on Rewriting Techniques and Applications*, volume 5595 of *LNCS*, pages 295–304, 2009. doi: 10.1007/978-3-642-02348-4_21.

[67] L. Kovács, G. Moser, and A. Voronkov. On transfinite Knuth–Bendix orders. In *Proc. 23rd International Conference on Automated Deduction*, volume 6803 of *LNCS*, pages 384–399, 2011. doi: 10.1007/978-3-642-22438-6_29.

[68] K. Kusakari, M. Nakamura, and Y. Toyama. Elimination transformations for associative-commutative rewriting systems. *Journal of Automated Reasoning*, 37(3):205–229, 2006. doi: 10.1007/S10817-006-9053-Y.

[69] D. Lankford. Canonical algebraic simplification in computational logic. Technical Report ATP-25, University of Texas, 1975.

[70] I. Lepper. Derivation lengths and order types of Knuth–Bendix orders. *Theoretical Computer Science*, 269(1-2):433–450, 2001. doi: 10.1016/S0304-3975(01)00015-9.

[71] I. Lepper. Simply terminating rewrite systems with long derivations. *Archive for Mathematical Logic*, 43(1):1–18, 2004. doi: 10.1007/S00153-003-0190-2.

[72] B. Löchner. Things to know when implementing KBO. *Journal of Automated Reasoning*, 36(4):289–310, 2006. doi: 10.1007/S10817-006-9031-4.

[73] B. Löchner. Things to know when implementing LPO. *International Journal on Artificial Intelligence Tools*, 15(1):53–80, 2006. doi: 10.1142/S0218213006002564.

[74] S. Lucas and R. Gutiérrez. Use of logical models for proving infeasibility in term rewriting. *Information Processing Letter*, 136, 2018. doi: 10.1016/J.IPL.2018.04.002.

[75] M. Ludwig and U. Waldmann. An extension of the knuth-bendix ordering with lpo-like properties. In *Proc. 14th International Conference on Logic Programming and Automated Reasoning*, volume 4790 of *LNCS*, pages 348–362, 2007. doi: 10.1007/978-3-540-75560-9_26.

[76] U. Martin and T. Nipkow. Ordered rewriting and confluence. In *Proc. 10th International Conference on Automated Deduction*, volume 449 of *LNCS*, pages 366–380, 1990. doi: 10.1007/3-540-52885-7_100.

[77] A. Middeldorp. Approximating dependency graphs using tree automata techniques. In *Proc. 1st International Joint Conference on Automated Reasoning*, volume 2083 of *LNCS*, pages 593–610, 2001. doi: 10.1007/3-540-45744-5_49.

[78] A. Middeldorp and H. Zantema. Simple termination of rewrite systems. *Theoretical Computer Science*, 175:127–158, 1997. doi: 10.1016/S0304-3975(96)00172-7.

[79] A. Middeldorp, J. Nagele, and K. Shintani. CoCo 2019: report on the eighth confluence competition. *International Journal on Software Tools for Technology Transfer*, 23(6):905–916, 2021. doi: 10.1007/S10009-021-00620-4.

[80] F. Mitterwallner, A. Middeldorp, and R. Thiemann. Linear termination is undecidable. In *Proc. 39th Symposium on Logic in Computer Science*, pages 57:1–57:12, 2024. doi: 10.1145/3661814.3662081.

[81] G. Moser. Derivational complexity of Knuth–Bendix orders revisited. In *Proc. 13th International Conference on Logic Programming and Automated Reasoning*, volume 4246 of *LNCS*, pages 75–89, 2006. doi: 10.1007/11916277_6.

[82] R. Nieuwenhuis. Simple LPO constraint solving methods. *Information Processing Letter*, 47(2):65–69, 1993. doi: 10.1016/0020-0190(93)90226-Y.

[83] T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002. doi: 10.1007/3-540-45949-9.

[84] H. Ochiai, T. Aoto, and Y. Toyama. AC-termination by modified monotonic AC-compatible semantic path orderings. In *COMP2004-76*, pages 23–31, March 2005.

[85] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer, 2002. doi: 10.1007/978-1-4757-3661-8.

[86] A. Rubio. A fully syntactic AC-RPO. *Information and Computation*, 178: 515–533, 2002. doi: 10.1006/inco.2002.3158.

[87] T. Saito and N. Hirokawa. Weighted path orders are semantic path orders. In *Proc. 14th International Symposium on Frontiers of Combining Systems*, volume 14279 of *LNCS (LNAI)*, 2023. doi: 10.1007/978-3-031-43369-6_4.

[88] T. Saito and N. Hirokawa. Lexicographic combination of reduction pairs. In *Proc. 30th International Conference on Automated Deduction*, LNCS (LNAI), 2025. To appear.

[89] J. Schöpf. The weighted path order in TTT2. Master's thesis, University of Innsbruck, 2020.

[90] J. Steinbach. Generating polynomial orderings. *Information Processing Letter*, 49(2):85–93, 1994. doi: 10.1016/0020-0190(94)90032-9.

[91] J. Steinbach. Simplification orderings: History of results. *Fundamenta Informaticae*, 24(1/2):47–87, 1995. doi: 10.3233/FI-1995-24123.

[92] C. Sternagel and R. Thiemann. Certified subterm criterion and certified usable rules. In *Proc. 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *LIPIcs*, pages 325–340, 2010. doi: 10.4230/LIPIcs.RTA.2010.325.

[93] C. Sternagel and R. Thiemann. Formalizing Knuth–Bendix orders and Knuth–Bendix completion. In *Proc. 13th International Conference on Rewriting Techniques and Applications*, volume 21 of *LIPIcs*, pages 287–302, 2013. doi: 10.4230/LIPIcs.RTA.2013287.

[94] C. Sternagel and A. Yamada. Reachability analysis for termination and confluence of rewriting. In *Proc. 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 11427 of *LNCS*, page 262–278, 2019. doi: 10.1007/978-3-030-17462-0_15.

[95] C. Sternagel, R. Thiemann, and A. Yamada. A formalization of weighted path orders and recursive path orders. *Archive of Formal Proofs*, September 2021. https://isa-afp.org/entries/Weighted_Path_Order.html, Formal proof development.

[96] A. Stump and B. Löchner. Knuth–Bendix completion of theories of commuting group endomorphisms. *Information Processing Letter*, 98:195–198, 2006. doi: 10.1016/j.ipl.2006.01.009.

[97] Terese. *Term Rewriting Systems*. Cambridge University Press, 2003.

[98] Termination Community. The Termination Problem Database (TPDB). https://github.com/TermCOMP/TPDB-ARI. Accessed: 05.11.2025.

[99] R. Thiemann. *The DP Framework for Proving Termination of Term Rewriting*. PhD thesis, RWTH Aachen University, 2007.

[100] R. Thiemann and C. Sternagel. Certification of termination proofs using CeTA. In *Proc. 22nd International Conference on Theorem Proving in Higher Order Logics*, volume 5674 of *LNCS*, pages 452–468, 2009. doi: 10.1007/978-3-642-03359-9_31.

[101] R. Thiemann, J. Giesl, and P. Schneider-Kamp. Improved modular termination proofs using dependency pairs. In *Proc. 2nd International Joint Conference on Automated Reasoning*, volume 3097 of *LNCS (LNAI)*, pages 75–90, 2004. doi: 10.1007/978-3-540-25984-8_4.

[102] R. Thiemann, G. Allais, and J. Nagele. On the Formalization of Termination Techniques based on Multiset Orderings. In *Proc. 23rd International Conference on Rewriting Techniques and Applications*, volume 15 of *LIPIcs*, pages 339–354, 2012. doi: 10.4230/LIPIcs.RTA.2012.339.

[103] R. Thiemann, J. Schöpf, C. Sternagel, and A. Yamada. Certifying the weighted path order (invited talk). In Zena M. Ariola, editor, *Proc. 5th International Conference on Formal Structures on Computation and Deduction*, volume 167 of *LIPIcs*, pages 4:1–4:20, 2020. doi: 10.4230/LIPICS.FSCD.2020.4.

[104] Y. Toyama. Termination of S-expression rewriting systems: Lexicographic path ordering for higher-order terms. In *Proc. 15th International Conference on Rewriting Techniques and Applications*, volume 3091 of *LNCS*, pages 40–54, 2004. doi: 10.1007/978-3-540-25979-4_3.

[105] X. Urbain. Modular & incremental automated termination proofs. *Journal of Automated Reasoning*, 34(4):315–355, 2004. doi: 10.1007/BF03177743.

[106] I. Wehrman and A. Stump. Mining propositional simplification proofs for small validating clauses. *Electronic Notes in Theoretical Computer Science*, 144 (2):79–91, 2005. doi: 10.1016/J.ENTCS.2005.12.008. Proc. 3rd Workshop on Pragmatics of Decision Procedures in Automated Reasoning.

[107] I. Wehrman, A. Stump, and E. M. Westbrook. Slothrop: Knuth–Bendix completion with a modern termination checker. In *Proc. 17th International Conference on Rewriting Techniques and Applications*, volume 4098 of *LNCS*, pages 287–296, 2006. doi: 10.1007/11805618_22.

[108] A. Weiermann. Complexity bounds for some finite forms of Kruskal's theorem. *Journal of Symbolic Computation*, 18(5):463–488, 1994. doi: 10.1006/JSCO.1994.1059.

[109] A. Weiermann. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths.

Theoretical Computer Science, 139(1&2):355–362, 1995. doi: 10.1016/0304-3975(94)00135-6.

[110] S. Winkler. A ground joinability criterion for ordered completion. In *Proc. 6th Workshop on Confluence*, pages 45–49, 2017. URL https://profs.scienze.univr.it/~winkler/papers/W-IWC17.pdf.

[111] S. Winkler, H. Zankl, and A. Middeldorp. Ordinals and knuth-bendix orders. In *Proc. 18th International Conference on Logic Programming and Automated Reasoning*, volume 7180 of *LNCS*, pages 420–434, 2012. doi: 10.1007/978-3-642-28717-6_33.

[112] S. Winkler, H. Sato, M. Kurihara, and A. Middeldorp. Multi-completion with termination tools (system description). *Journal of Automated Reasoning*, 50:317–354, 2013. doi: 10.1007/978-3-540-71070-7_26.

[113] A. Yamada. *The Weighted Path Order for Termination of Term Rewriting*. PhD thesis, Nagoya University, 2014.

[114] A. Yamada. Term orderings for non-reachability of (conditional) rewriting. In *Proc. 11th International Joint Conference on Automated Reasoning*, volume 13385 of *LNCS*, pages 248–267, 2022. doi: 10.1007/978-3-031-10769-6_15.

[115] A. Yamada. Tuple interpretations for termination of term rewriting. *Journal of Automated Reasoning*, 66(4):667–688, 2022. doi: 10.1007/S10817-022-09640-4.

[116] A. Yamada, K. Kusakari, and T. Sakabe. Nagoya Termination Tool. In *Proc. Joint 25th International Conference on Rewriting Techniques and Applications and 12th International Conference on Typed Lambda Calculi and Applications*, volume 8560, pages 466–475, 2014. doi: 10.1007/978-3-319-08918-8_32.

[117] A. Yamada, K. Kusakari, and T. Sakabe. A unified ordering for termination proving. *Science of Computer Programming*, 111:110–134, 2015. doi: 10.1016/j.scico.2014.07.009.

[118] A. Yamada, S. Winkler, N. Hirokawa, and A. Middeldorp. AC-KBO revisited. *Theory Pract. Log. Program.*, 16(2):163–188, 2016. doi: 10.1017/S1471068415000083.

[119] H. Zankl, N. Hirokawa, and A. Middeldorp. KBO orientability. *Journal of Automated Reasoning*, 43:173–201, 2009. doi: 10.1007/s10817-009-9131-z.

[120] H. Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17:23–50, 1994. doi: 10.1006/jsco.1994.1003.

[121] H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995. doi: 10.3233/FI-1995-24124.

Index

Publications

in peer-reviewed international conferences

(i) Teppei Saito and Nao Hirokawa: "Lexicographic Combination of Reduction Pairs". *Proceedings of the 30th International Conference on Automated Deduction*, to appear in *Lecture Notes in Artificial Intelligence*, 2025.

(ii) Dohan Kim, Teppei Saito, René Thiemann and Akihisa Yamada: "An Isabelle Formalization of Co-rewrite Pairs for Non-reachability in Term Rewriting". *Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 272–282, 2025.

(iii) Teppei Saito and Nao Hirokawa: "Simulating Dependency Pairs by Semantic Labeling". *Proceedings of the 9th International Conference on Formal Structures for Computation and Deduction*, volume 299 of *Leibniz International Proceedings in Informatics*, pages 13:1–13:20, 2024.

(iv) Teppei Saito and Nao Hirokawa: "Weighted Path Orders are Semantic Path Orders". *Proceedings of the 14th International Symposium on Frontiers of Combining Systems*, volume 14279 of *Lecture Notes in Artificial Intelligence*, pages 63–80, 2023.