Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT

Vademecum of Polynomial Orderings

J. Steinbach & M. Zehnter

SEKI Report SR-90-03

Vade-mecum of Polynomial Orderings

Joachim Steinbach
Michael Zehnter

Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
D-6750 Kaiserslautern
F.R.G.
e-mail: steinba@informatik.uni-kl.de

This research was supported by the Deutsche Forschungsgemeinschaft, SFB 314 (D4)

Abstract

Orderings on polynomial interpretations of operators represent a powerful technique for proving the termination of rewrite systems. We discuss most of the well-known features concerning this method including heuristics for the generation of adequate interpretations orienting a given system, strategies for deciding the positiveness of a polynomial, restricted interpretations to guarantee E-termination and some improvements (such as the Cartesian product of polynomials). This report is a detailed source of examples that illustrate the uncertainties of polynomial orderings. Furthermore, we develop new aspects involving two approaches that are suitable to simple implementations, a few suggestions for producing adequate interpretations (e.g., a technique for automatically generating linear polynomials), some new orderings handling underlying theories, a new improved version of polynomial orderings (based on the Knuth-Bendix ordering with status) as well as a detailed comparison of the various approaches together with path and decomposition orderings. Thus, the report in hand is a kind of vade-mecum of most of the known as well as new results concerned with polynomial orderings.

Keywords E-compatible, E-termination, Homeomorphic embedding, Interpretation, Knuth-Bendix ordering, Lexicographical ordering, Monomial, Multiset ordering, Path and decomposition orderings, Polynomial, Simplification ordering, Status, Termination, Term rewriting system, Well-founded ordering

Contents

1 Introduction and Notation

Polynomial orderings represent an approach for proving the termination of term rewriting systems. Dealing with so-called termination orderings presumes a familiarity with the basic concepts of term rewriting systems as well as the special notation used on this subject. In the following pages we give a brief summary of what term rewriting systems are and introduce the notations that are essential for the remainder of this paper.

1.1 Motivation

Term rewriting systems provide a powerful tool for expressing non-deterministic computations and as a result they have been widely used in formula-manipulation and theorem-proving systems. Moreover, there exists the potential for their application in many other areas of computer science and mathematics such as abstract data type specifications and program verification.

As programs they have a very simple syntax and their semantic is based on equalities that are used as reduction rules with no explicit control. For this it is essential that term rewriting systems have two main properties: confluence and termination. A non-confluent term rewriting system may sometimes be transformed into a confluent one using the Knuth-Bendix completion procedure ([KB70]). However, termination is the crucial point in the realization of this algorithm which tests the local confluence and requires termination for inferring the total confluence.

There exist various methods for proving termination of rewrite systems. Most of them have in common the main idea of embedding the reduction relation $\Longrightarrow_{\mathfrak{R}}$ (induced by the rule system $\mathfrak{R}$) into a well-founded ordering >. To check the inclusion '$\Longrightarrow_{\mathfrak{R}} \subseteq >$' all infinitely possible derivations must be tested. The key idea is to restrict this infinite test to a finite one by requiring > to be a reduction ordering. This means that, in addition to the well-foundedness, > must have the replacement property (also called compatibility with the structure of terms) and the stability by instantiations.

One way of constructing reduction orderings was proposed by the authors (Knuth and Bendix) of the completion procedure ([KB70]) and so are called Knuth-Bendix orderings. They joined two general principles each of which has been extended in the following years. Firstly, the idea of giving an ordering on the set of function symbols and comparing terms by comparing the function symbols they are composed of, led to the so-called path orderings that realize a more syntactical comparison of terms.

The second component of Knuth-Bendix orderings is a mapping of the function symbols to the naturals (a so-called weight function or termination function) that is extendable to terms by adding the weights of the function symbols they contain. By modifying this termination function such that the image of the set of function symbols is a subset of all polynomials over the reals, one virtually gets the polynomial orderings ([La79]). Contrary to path orderings (and Knuth-Bendix orderings) polynomial orderings are not based on precedence orderings over the function symbols.

The object of this paper is to give a survey of both well-known and new facts connected with proving termination using polynomial orderings. In the next chapter we introduce a general definition of polynomial orderings. Following this theoretical result we show how to get easily implementable, restricted versions of this definition that unify all implementations known to us. Several well-known heuristics for finding well-suited interpretations for certain function symbols are collected in chapter 3 ([Be86], [BL87]). Moreover, we present some new heuristics as well as a test for the complexity that polynomials, interpreting a certain function symbol, must have. Finally, we describe an algorithm based on the Simplex method for finding an interpretation to a given rule system provided this system can be oriented using linear polynomials.

The use of polynomial orderings as reduction orderings in the completion procedure requires methods of proving the positiveness of polynomials. In chapter 4 two of these well-known methods are illustrated ([BL87], [Ro88]). Chapter 5 deals with polynomial orderings that are suitable to prove the termination of term rewriting systems based on underlying equational theories. In addition to well-known orderings ([Be86]), some new ones are presented here.

Taking a pattern by the Knuth-Bendix orderings with status it is possible to extend a polynomial ordering by introducing a precedence ordering on the set of function symbols. This new approach called improved polynomial ordering (and some other well-known ones, [La79]) is introduced in chapter 6. Finally, a conclusive comparison of the various kinds of polynomial orderings presented, including path and decomposition orderings, as well as the Knuth-Bendix ordering will be given.

1.2 Notation

Here we briefly recapitulate the most essential notions, used in connection with term rewriting systems, that are found in this paper. The reader can find a more detailed presentation in [AM89], [HO80] or [JL87].

Let $\mathfrak{F}$ be a set of function symbols possessing fixed arity and $\mathfrak{X}$ some denumerable infinite set of variables, then Γ denotes the set of all terms over $\mathfrak{F}$ and $\mathfrak{X}$ whereas Γ_G is the set of all ground terms over $\mathfrak{F}$. The set of all variables of a term t will be denoted by V(t). The leading function symbol and the tuple of the (direct) arguments of a term t are referred to by top(t) and args(t), respectively. The empty term will be denoted by λ.

A substitution σ is defined as an endomorphism on Γ with the finite domain $\{x \mid \sigma(x) \neq x\}$, i.e. σ simultaneously replaces all variables of a term by terms. We use the formalism of positions of terms which are sequences of non-negative integers. The empty sequence is denoted by ε. The set of all positions of a term t is called the set of occurrences and its abbreviation is O(t). We write $t[u \leftarrow s]$ to denote the term that results from t by replacing t/u (the subterm of t at occurrence $u \in O(t)$) by s.

A term rewriting system (TRS, for short) $\mathfrak{R}$ over Γ is a finite or countably infinite set of rules each of the form $l \rightarrow_{\mathfrak{R}} r$, where l and r are terms in Γ such that every variable that occurs in r also occurs in l.

The binary relation generated by a TRS $\mathfrak{R}$ on Γ is named $\Longrightarrow_{\mathfrak{R}}$. The transitive closure of $\Longrightarrow_{\mathfrak{R}}$ is denoted by $\overset{+}{\Longrightarrow}_{\mathfrak{R}}$ whereas the transitive and reflexive closure of $\Longrightarrow_{\mathfrak{R}}$ is denoted by $\overset{*}{\Longrightarrow}_{\mathfrak{R}}$.

A binary relation > on a set $\mathfrak{M}$ is said to be well-founded if there are no infinite descending chains of the form $m_1 > m_2 > \dots$ with $m_i \in \mathfrak{M}$. Analogous with $\mathfrak{M}$, a TRS $\mathfrak{R}$ is well-founded if there exists no sequence (called derivation) $t_0 \Longrightarrow_{\mathfrak{R}} t_1 \Longrightarrow_{\mathfrak{R}} t_2 \Longrightarrow_{\mathfrak{R}} \dots$ that is infinitely descending.

A partial ordering is a transitive and irreflexive binary relation. A partial ordering $\triangleright$ on $\mathfrak{F}$ is called a precedence and a partial ordering on Γ is called a term ordering.

An example for a partial ordering on terms is the irreflexive part $\sqsupset$ of the so-called homeomorphic embedding relation $\sqsupseteq$ that is defined by

$$s = f(s_1,\dots,s_m) \sqsupseteq g(t_1,\dots,t_n) = t \quad \text{iff} \quad (1)\ f = g \text{ and } s_i \sqsupseteq t_i \text{ for all } i \in \{1,\dots,n\}$$
$$\text{or} \quad (2)\ s_i \sqsupseteq t \text{ for at least one } i \in \{1,\dots,n\}$$

We will use this simplified version of $\sqsupseteq$ since we are only faced with function symbols having fixed arity, here. The main point of the homeomorphic embedding relation is the fact that a partial ordering > is well-founded if it contains $\sqsupseteq$ ([De82]).

On the other hand, we have that each non-terminating (i.e. non-well-founded) partial ordering is self-embedding ([De82]). That means that each infinite descending chain $t_1 > t_2 > \ldots$ contains t_i, t_j with $i < j$ and $t_j \sqsupset t_i$.

As previously noted, a reduction ordering > is a well-founded and stable (by instantiations) partial ordering that is compatible with the structure of terms (it has the replacement property). That means, $t_1 > t_2$ implies $t[u \leftarrow t_1] > t[u \leftarrow t_2]$ for any $t, t_1, t_2 \in \Gamma$ and $u \in O(t)$. In other words, decreasing a subterm decreases any superterm containing it.

One special class of reduction orderings is that of the simplification orderings ([De82]). The key idea of the concept of simplification orderings is to guarantee the well-foundedness of an ordering > by requiring the so-called subterm property (which in turn ensures that $\sqsupset$ is contained in >). An ordering > on terms is said to have the subterm property if for every term t and for every function symbol f, $f(\ldots,t,\ldots) > t$ holds. Since every simplification ordering is well-founded, every simplification ordering is a reduction ordering.

The main advantage of introducing simplification orderings is that the test for the subterm property provides a very handy criterion for the test of well-foundedness.

2 An Approach to Polynomial Orderings

In this chapter we investigate whether the image of a polynomial interpretation must be a well-founded subset of ℝ or the interpreting polynomials must have a kind of subterm property inducing the corresponding ordering to be a simplification ordering. We introduce a formal definition of polynomial orderings which enables us to state whether a polynomial is appropriate as an interpretation for a function symbol.

In addition to this (theoretical) result we propose a more restrictive version of polynomial orderings that generalizes those given in [De87], [La79], [BL87], [Be86], [Pa89], [Ze89] and [MN70]. It provides a simple technique for implementing polynomial orderings.

2.1 Definitions

Let $r_1,\dots,r_n \in \mathbb{N}_o$ (= $\mathbb{N} \cup \{0\}$) and $c \in \mathbb{R}$. A monomial in $\mathbb{R}^n$ of degree $r = r_1 + \dots + r_n$ is a function $m : \mathbb{R}^n \to \mathbb{R}$ with $(X_1,\dots,X_n) \to cX_1^{r_1}X_2^{r_2} \dots X_n^{r_n}$. A polynomial of degree r over $\mathbb{R}^n$ is a sum of monomials of degree less than or equal to r with at least one monomial having degree r. This means that a polynomial p of degree r is of the form

$$p : \mathbb{R}^n \to \mathbb{R}$$
$$p(X_1,\dots,X_n) = \sum_{r_1,\dots,r_n \in \mathbb{N}_o} c_{r_1 \dots r_n} X_1^{r_1} \dots X_n^{r_n} \quad \text{with } r_1+\dots+r_n \leq r.$$

Thus, $m(X,Y,Z) = 5X^2YZ^4$ is a monomial in $\mathbb{R}^3$ of degree 7 and $p(X,Y,Z) = 3X^2Z^2 + Y - 4X^4Y$ is a polynomial in $\mathbb{R}^3$ of degree 5.

A polynomial p will have strict arity n if there occur n variables in p that differ by pairs.

The set of all polynomials $p : \mathbb{R}^n \to \mathbb{R}$ with coefficients $c_i \in \mathbb{R}$ is denoted by $\mathfrak{Pol}(\mathbb{R})$. A polynomial $p : \mathbb{R}^n \to \mathbb{R}$ whose n variables are all instantiated by real numbers is called a ground polynomial or constant polynomial. For example, $3 * 1.5^2 * 2^2 + (-1) - 4 * 1.5^4 * (-1)$ is the ground polynomial one obtains by substituting the variables (X,Y,Z) of the previous polynomial by (1.5 , -1 , 2).

Since every ground polynomial is equal to a real number we will identify the set of ground polynomials with $\mathbb{R}$. The set of variables over $\mathbb{R}$ that occur in polynomials of $\mathfrak{Pol}(\mathbb{R})$ is named $\mathfrak{V}$. To distinguish variables over Γ_G and variables over $\mathbb{R}$ we use capital letters for elements of $\mathfrak{V}$.

It is apparent that the relations between $\mathfrak{Pol}(\mathbb{R})$, $\mathbb{R}$ and $\mathfrak{V}$ are similar to the relations between Γ, Γ_G and $\mathfrak{X}$.

2.2 Polynomial Orderings on Ground Terms

Polynomial orderings have been studied by Manna/Ness ([MN70]), Lankford ([La75], [La79]), Dershowitz ([De83], [De87]) and Ben Cherifa/Lescanne ([Be86], [BL85], [BL86], [BL87]). Manna/Ness, Lankford and Ben Cherifa/Lescanne proposed a method which maps the set of terms to a well-founded set by attaching monotonous functions to operators. Dershowitz's technique uses an arbitrary ordered set by requiring the monotonous functions to possess a kind of subterm property ([De79]). In this chapter we present an approach similar to that of Dershowitz. The main difference to his technique is the fact that we use a weaker version of the subterm property by allowing identity functions as interpretations.

In order to get an ordering > on ground terms, we map these terms to $\mathbb{R}$ by a so-called polynomial interpretation $[\ldots]$. On $\mathbb{R}$ we have the natural ordering >.

Definition 2.2.1

A polynomial interpretation for ground terms $[\ldots]: \mathfrak{F} \rightarrow \mathfrak{Pol}(\mathbb{R})$ maps each n-ary function symbol $f \in \mathfrak{F}$ to a polynomial $p \in \mathfrak{Pol}(\mathbb{R})$ with strict arity n. Each polynomial interpretation $[\ldots]$ for ground terms can be extended to $[\ldots] : \Gamma_G \rightarrow \mathbb{R}$ by $[f(t_1,\ldots,t_n)] = [f]([t_1],\ldots,[t_n])$ defining an ordering $>_{POL}$ on Γ_G by

$$s >_{POL} t \quad \text{iff} \quad [s] > [t]$$

where > is the usual ordering on $\mathbb{R}$. On Γ_G there is an equivalence relation $=_{POL}$ defined by $s =_{POL} t$ iff $[s] = [t]$. ∎

As usual, $s \geq_{POL} t$ means $s >_{POL} t$ or $s =_{POL} t$.

Note that the transfer of the total ordering > on $\mathbb{R}$ to Γ_G as by defining $s >_{POL} t$ iff $[s] > [t]$ is not total. Let, for example, $[\ldots]$ be defined as $[a]() = 2$, $[f](x,y) = xy$ and $[g](x,y) = x+y$. Then, we have $[g(a,a)] = 4 = [f(a,a)]$. Thus, $g(a,a)$ and $f(a,a)$ are incomparable w.r.t. $>_{POL}$.

We will denote the set of polynomials that are attached to any $f \in \mathfrak{F}$ by a polynomial interpretation $[\ldots]$ by $[\mathfrak{F}]$. Note that $[\mathfrak{F}]$ is a proper and finite (since $\mathfrak{F}$ is finite) subset of $\mathfrak{Pol}(\mathbb{R})$. According to $[\mathfrak{F}]$, $[\Gamma_G]$ is the set of real numbers that are the image of any ground term under $[\ldots]$.

Since we are interested in reduction orderings on terms we need to check the compatibility with the structure of terms of $>_{POL}$ on the one hand and the well-foundedness on the other hand. Let us first discuss the compatibility. The use of polynomials defining orderings on ground terms requires the monotony of the interpreting polynomials to guarantee the compatibility of the corresponding ordering $>_{POL}$. From analysis we know that a function, and especially a polynomial, is monotonous if all its first derivatives are positive. This means that the first derivative $dp(X_1,...,X_n)/dX_i$ for each variable X_i $(1 \leq i \leq n)$ occurring in p is positive for each instantiation of the X_i to real numbers. However, do we really need the monotony over the whole interval $\mathbb{R}$? The following example provides an answer to this issue.

Example 2.2.2

Let
$$[0]() = 2$$
$$[s](x) = X^2$$
$$[+](x,y) = X^2Y - X$$

This interpretation defines an ordering $>_{POL}$ on Γ_G that is compatible with the structure of terms, although, $X^2Y - X$ is not monotonous since the first derivative for X at point $(X,Y) = (0.5, 0.5)$, $d(X^2Y - X)/dX = 2XY - 1 = -0.5$, is negative.
This is derived from the fact that $[t] \geq 2$ for all terms $t \in \Gamma_G$. $X^2Y - X$ is monotonous for all instantiations of X and Y by real numbers greater than or equal to 2 since the partial derivatives of $X^2Y - X$ ($2XY-1$ and X^2) are positive if $X,Y \geq 2$. ∎

By analyzing these facts we can weaken the request for the monotony of the interpreting polynomials in the following way: Each polynomial has to be monotonous only for instantiations which map the variables to real numbers contained in $[\Gamma_G]$.

In example 2.2.2, $[\Gamma_G]$ is $\{2,4,6,14,16,...\}$. We have to guarantee that, for example, $X^2Y - X$ at $(4,4)$ is greater than $X^2Y - X$ at $(4,2)$. But we do not have to prove that it is greater at $(4,4)$ than at $(4,3)$ since $3 \notin [\Gamma_G]$. This property is called monotony w.r.t. $[\Gamma_G]$. The following lemma shows that this criterion is both necessary as well as sufficient.

Lemma 2.2.3

Let [...] be a polynomial interpretation for ground terms.
Then, the corresponding ordering $>_{POL}$ on Γ_G is compatible with the structure of terms if and only if each polynomial $p \in [\mathfrak{F}]$ is monotonous w.r.t. $[\Gamma_G]$. ∎

The most difficult problem arising from this lemma is that of deciding whether a certain polynomial is monotonous w.r.t. $[\Gamma_G]$. It is the whole set $[\mathfrak{F}]$ of polynomials attached by $[...]$ to the set of function symbols $\mathfrak{F}$ that we have to take into account. Particularly, the values attached to the constants have a significant influence on our decision.

More precisely, we first have to define $[...]$ by constituting the polynomials of each function symbol. Secondly, the fixation of $[\Gamma_G]$ will describe the set of reals that could be the image of any ground term. Finally, we have to verify that each polynomial in $[\mathfrak{F}]$ is monotonous w.r.t. $[\Gamma_G]$.

Computing $[\Gamma_G]$ as well as proving a polynomial to be monotonous on a subset of $\mathbb{R}$ can be rather tedious. Moreover, since this problem generally invokes proving a polynomial with domain $D \subset \mathbb{R}$ is greater than 0, it is in general undecidable (for more details about this point, see chapter 4).

In summary, it can be stated that the criterion given in lemma 2.2.3 for the compatibility of $>_{POL}$ is not particularly useful and so we need to develop a more appropriate check.

As a result we will restrict the set of potentially interpreting polynomials from $\mathfrak{Pol}(\mathbb{R})$ to $\mathfrak{Pol}(\mathbb{R}^+)$, the set of polynomials that have positive real coefficients. This restriction guarantees that each ground term is mapped to a real number greater than zero (see lemma 2.2.4). So, $[\Gamma_G]$ is implicitly computed to be a subset of $\mathbb{R}^+$ when choosing the interpreting polynomials.

Lemma 2.2.4 ([Ze89])

Let $[...]$ be a polynomial interpretation for ground terms that attaches a strict n-ary polynomial $p \in \mathfrak{Pol}(\mathbb{R}^+)$ to each n-ary function symbol $f \in \mathfrak{F}$. Then,

$$[t] > 0 \text{ for all terms } t \in \Gamma_G. \qquad \blacksquare$$

From this fact and since each polynomial $p \in \mathfrak{Pol}(\mathbb{R}^+)$ will be monotonous if we restrict its domain to $\mathbb{R}^+$, the following lemma holds.

Lemma 2.2.5 ([Ze89])

Let $[...]$ be a polynomial interpretation for ground terms that attaches a strict n-ary polynomial $p \in \mathfrak{Pol}(\mathbb{R}^+)$ to each n-ary function symbol $f \in \mathfrak{F}$. Then, the corresponding ordering $>_{POL}$ is a partial ordering on Γ_G being compatible with the structure of terms. ■

Example 2.2.6

Let
$$[0]() = 2$$
$$[s](x) = X^2$$
$$[+](x, y) = X^2Y$$

As a result of lemma 2.2.4, $[t] > 0$ for all $t \in \Gamma_G$. Therefore, the corresponding ordering $>_{POL}$ is a partial ordering on Γ_G being compatible with the structure of terms. ∎

Note that the requirement of the coefficients being positive is a significant restriction. However, a less restrictive characterization for monotonous interpretations would be too complex and intractable for unfamiliar users.

Let us now consider the termination property. Our approach is similar to that of guaranteeing the compatibility. Firstly, we will describe the set of polynomials in question for a noetherian polynomial ordering on terms that is compatible. This means that we have to give sufficient and necessary constraints for these polynomials. Analogous with the compatibility, this most general criterion is quite unhandy. Therefore, we will give a characterization of a class of polynomials leading to reduction orderings that is of great importance for practical application since the user is able to determine directly whether a polynomial belongs to this class. The decisions in this connection are based on the following theorem.

Theorem 2.2.7 ([De82])

Any total monotonous ordering $>$ on Γ_G over a finite set $\mathfrak{F}$ of fixed-arity function symbols is well-founded if and only if it possesses the subterm property. ∎

Since we consider here monotonous orderings on Γ_G over finite sets $\mathfrak{F}$ of fixed-arity function symbols, this theorem is of significant importance for our analysis of polynomial orderings. The reason why we were unable to directly make use of theorem 2.2.7 is that polynomial orderings, as noted previously, are not total on Γ_G. Thus, we have to adapt it to our necessities.

Suppose there exists a polynomial $p \in [\mathfrak{F}]$ such that $t_i > p[\dots,t_i,\dots]$ for any t_i. Since p is monotonous w.r.t. $[\Gamma_G]$ (to guarantee the compatibility of $>_{POL}$) it holds that $p[\dots,t_i,\dots] > p[\dots,p[\dots,t_i,\dots],\dots] > \dots$ and so $>_{POL}$ is not well-founded. Thus, we have to exclude so-called diminishing polynomials. In detail, we will only need the property of being non-diminishing on $[\Gamma_G]$.

Definition 2.2.8

A polynomial $p[X_1,\ldots,X_n]$ is non-diminishing on $D \subset \mathbb{R}$ if for all $i \in \{1,\ldots,n\}$ and for all $t_i \in D$ it holds that $p[\ldots,t_i,\ldots] \geq t_i$. ∎

Theorem 2.2.9

Let $[\ldots]$ be a polynomial interpretation for ground terms.
Then, the corresponding ordering $>_{POL}$ on Γ_G is compatible with the structure of terms and well-founded if and only if each polynomial $p \in [\mathfrak{F}]$ is monotonous and non-diminishing on $[\Gamma_G]$. ∎

As done before when discussing the compatibility we cannot decide if a polynomial leads to a well-founded ordering $>_{POL}$ by just looking at this certain polynomial. Again, it is the whole set $[\mathfrak{F}]$ we have to take into account and even more this decision depends on the values attached to the constants since the smallest value interpreting a constant is the lower bound of $[\Gamma_G]$ (if the polynomials are monotonous and non-diminishing).

Defining a well-founded polynomial ordering that is compatible with the structure of terms consists of three steps:

(1) Fixing the polynomials to be attached to each function symbol,
(2) Computing $[\Gamma_G]$ and
(3) Verifying that all polynomials $p \in [\mathfrak{F}]$ are monotonous and non-diminishing on $[\Gamma_G]$.

It should be noted that there does not exist a characterization of the set of polynomials leading to well-founded and compatible polynomial orderings that is more general than that given in theorem 2.2.9.

Unfortunately, this characterization is not entirely convenient for computer implementation. However, we will now show that it is easy to deduce a more restrictive as well as feasible characterization from this theorem.

Lemma 2.2.10

Let $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$ be the set of all polynomials $p : \mathbb{R}^n \rightarrow \mathbb{R}$ of any arity $n \in \mathbb{N}_0$ that have coefficients of $[1, \infty]$. Let $[\ldots]$ be a polynomial interpretation for ground terms that attaches to each n-ary function symbol $f \in \mathfrak{F}$ a strict n-ary polynomial $p \in \mathfrak{Pol}(\mathbb{R}^{\geq 1})$.
Then, the corresponding ordering $>_{POL}$ is well-founded and compatible with the structure of terms. ∎

The proof is straightforward utilizing theorem 2.2.9. Note that the restriction to $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$ can easily be weakened by adding some additional constraints as shown in the following lemma.

Lemma 2.2.11

Let [...] be a polynomial interpretation for ground terms that attaches

(1) a value $C \in \mathbb{R}$ to each constant function symbol $c \in \mathfrak{F}$ such that $C \geq 1$

(2) a strict n-ary polynomial p to each n-ary (n > 0) function symbol $f \in \mathfrak{F}$ such that

$$p(X_1,\dots,X_n) = \sum_{r_1,\dots,r_n \in \mathbb{N}_0} c_{r_1 \dots r_n} X_1^{r_1} \dots X_n^{r_n}$$

such that for each i $(1 \leq i \leq n)$ $\sum_{\substack{r_i > 0 \\ r_1,\dots,r_n \in \mathbb{N}_0}} c_{r_1 \dots r_n} \geq 1$

Then, the corresponding ordering $>_{POL}$ on Γ_G is a reduction ordering. ∎

The condition (2) requires that the sum of the coefficients of those monomials containing X_i (with an exponent greater than 0) has to be greater than or equal to 1 for each variable X_i. For example, the polynomial $p(X,Y) = 0.5X^2Y + 0.5XY^2$ fulfils this condition.

The proof is also straightforward with theorem 2.2.9. Since all polynomials of $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$ fulfil the conditions above, this lemma is a strict generalization of lemma 2.2.10.

Note that, following lemma 2.2.11, only the coefficients of polynomials having an arity greater than 0 can potentially be less than 1.

In summary, for a definition of a reduction ordering on polynomial interpretations it is not necessary

(1) to restrict the domain of the polynomials to $\mathbb{N}$ or another well-founded subset of $\mathbb{R}$ or
(2) to restrict the subset of $\mathbb{R}$, the coefficients belong to, to $\mathbb{N}$ or
(3) to require the subterm property of the polynomials.

We only have to guarantee that the polynomials are

- monotonous w.r.t. $[\Gamma_G]$ and
- non-diminishing on $[\Gamma_G]$

whereby the similarity of the subterm property and the property 'non-diminishing' is obvious. In fact, each polynomial ordering having the subterm property is defined by non-diminishing polynomials but the converse does not hold.

Theorem 2.2.9 and even lemmata 2.2.10 and 2.2.11 generalize all versions of defining polynomial orderings we know ([La79], [Be86], [BL87], [Ze89], [Pa89], [De87], [MN70]).

2.3 Polynomial Orderings on Variable Terms

Working with the completion procedure requires reduction orderings on terms containing variables. In this section we give a formal definition of polynomial orderings on variable terms.

To lift polynomial orderings to variable terms we must extend definition 2.2.1 in such a way that it maps variable terms to variable polynomials.

Definition 2.3.1

A polynomial interpretation for variable terms $[\ldots] : \mathfrak{F} \rightarrow \mathfrak{Pol}(\mathbb{R})$
$\mathfrak{X} \rightarrow \mathfrak{V}$
maps each n-ary function symbol $f \in \mathfrak{F}$ to a polynomial $p \in \mathfrak{Pol}(\mathbb{R})$ of strict arity n and each variable $x \in \mathfrak{X}$ over terms to a variable $X \in \mathfrak{V}$ over $\mathbb{R}$.
This mapping can be extended to $[\ldots] : \Gamma \rightarrow \mathfrak{Pol}(\mathbb{R})$ by defining $[f(t_1,\ldots,t_n)] = [f]([t_1],\ldots,[t_n])$. ∎

We now show that it is possible to extend the orderings presented in section 2.2 to variable terms by using polynomial interpretations for variable terms instead of polynomial interpretations for ground terms. As a result we have to distinguish two kinds of ground substitutions: ground substitutions on terms like $\sigma_T : \mathfrak{X} \rightarrow \Gamma_G$ and ground substitutions on polynomials like $\sigma_P : \mathfrak{V} \rightarrow \mathbb{R}$. Ground substitutions on polynomials simply map variables over $\mathbb{R}$ to real numbers.

Firstly, we define an ordering on variable polynomials depending on the underlying polynomial interpretation.

Lemma 2.3.2 ([Ze89])

Let $[\ldots]$ be a polynomial interpretation of variable terms and $r,q \in \mathfrak{Pol}(\mathbb{R})$. By defining

$r >_P q$	iff	$\sigma_P(r) > \sigma_P(q)$	for all ground substitutions σ_P on polynomials with $\sigma_P(X) \in [\Gamma_G]$ for each variable X

we get an ordering $>_P$ on $[\Gamma] \subseteq \mathfrak{Pol}(\mathbb{R})$ that is stable by instantiations σ_P which guarantee $\sigma_P(X) \in [\Gamma_G]$ for all $X \in \mathfrak{V}$. ∎

We will call substitutions σ_P that guarantee $\sigma_P(X) \in [\Gamma_G]$ for all $X \in \mathfrak{V}$ $[\Gamma_G]$-substitutions.

Note that this definition of an ordering on variable polynomials takes into account that there is no need to compare all ground substitutions of the polynomials. For example, if [...] maps all ground terms t to real numbers greater than 2, it will not be necessary to guarantee the stability of $>_P$ against the ground substitution $\sigma_P : X \rightarrow -2$.

Now we are able to extend the ordering on ground terms defined in theorem 2.2.9 to variable terms.

Theorem 2.3.3

Let [...] be a polynomial interpretation for variable terms that maps $\mathfrak{F}$ to $\mathfrak{Pol}(\mathbb{R})$.
The ordering defined by

$$s >_{POL} t \quad \text{iff} \quad [s] >_P [t]$$

is a reduction ordering on Γ that is stable by instantiations if and only if each polynomial $p \in [\mathfrak{F}]$ is monotonous and non-diminishing on $[\Gamma_G]$.■

Comparing two terms s,t reduces to the comparison of two polynomials [s],[t] that are monotonous and non-diminishing on $[\Gamma_G]$ and this in turn reduces to comparing all $[\Gamma_G]$-substitutions $\sigma_P([s])$, $\sigma_P([t])$.

Thus, if we want to orient a certain set of rules by using a polynomial ordering we have to perform the following steps:

(1) Fixing the polynomials to be attached to each function symbol,
(2) Computing $[\Gamma_G]$,
(3) Verifying that all polynomials $p \in [\mathfrak{F}]$ are monotonous w.r.t. $[\Gamma_G]$,
(4) Verifying that all polynomials $p \in [\mathfrak{F}]$ are non-diminishing on $[\Gamma_G]$ and
(5) Proving that $\sigma_P([l]) > \sigma_P([r])$ for each rule $l \rightarrow r$ and each $[\Gamma_G]$-substitution σ_P.

In the following lemma we present a restricted version of theorem 2.3.3 that can more readily be applied (cf. section 2.2).

Lemma 2.3.4

Let [...] be a polynomial interpretation for variable terms that maps $\mathfrak{F}$ to $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$.

The ordering defined by

$$s >_{POL} t \quad \text{iff} \quad [s] >_P [t]$$

is a reduction ordering on Γ that is stable by instantiations. ∎

By using this lemma a given rule system can be oriented by merely

(1) finding appropriate polynomials and
(2) proving that $\sigma_P([l]) > \sigma_P([r])$ for each rule $l \to r$ and each $\mathbb{R}^{\geq 1}$-substitution σ_P.

The second step indicates that the problem of quantifying over all ground substitutions on terms is reduced to the problem of quantifying over certain ground substitutions on polynomials.

Note that this definition of polynomial orderings reveals the close connection between the interpretation of constant function symbols on the one side and the test of comparability of two terms, by inserting real numbers starting at a certain point S (here: S=1), on the other side. This starting point S has to be the smallest real number that is attached by [...] to a constant of $\mathfrak{F}$.

The reader should recognize that we are only interested in the smallest value of $[\Gamma_G]$ and that we wish to test the comparability of two polynomials by comparing all $\mathbb{R}^{\geq 1}$-substitutions instead of all $[\Gamma_G]$-substitutions, although $[\Gamma_G] \subset \mathbb{R}^{\geq 1}$. Since the methods of comparing polynomials are based on analytical arguments this difference does not matter. This even holds for $[\Gamma_G] \subseteq \mathbb{N}$. For more details about how to compare two polynomials, see chapter 4.

2.4 Examples

In order to illustrate the power as well as the difficulty encountered in the handling of polynomial orderings we will demonstrate several examples.

The polynomial ordering as presented in this chapter is implemented as part of the COMTES-system ([AMS89], [St89c], [Pa89], [AGGMS87]). COMTES is an extended Knuth-Bendix completion procedure and can be viewed as a parametric system that is particularly suited for efficiency experiments. Besides different reduction strategies and critical pair criteria, the parameters also include various termination methods.

The following examples have been tested by COMTES. Each of them contains an initial specification $\mathfrak{R}$ and an interpretation (if possible) which causes the termination of $\mathfrak{R}$. Moreover, the interpretation is sufficient for completing $\mathfrak{R}$ and thus it computes a canonical system (that is not explicitly given here).

Let us note that in the remaining part of this report, we no longer differentiate between capital letters for denoting variables of polynomials and small letters for representing variables of terms. *Henceforth, we will only use lower cases.*

Example 2.4.1 Loops

$$\mathfrak{R}: \begin{array}{lcl} x * (x\backslash y) & \to & y \\ (x/y) * y & \to & x \\ 1 * x & \to & x \\ x * 1 & \to & x \end{array}$$

$$\begin{array}{lcl} [1]() & = & 1 \\ [\backslash](x,y) & = & x + y \\ [/](x,y) & = & x + y \\ [*](x,y) & = & x + y \end{array}$$

∎

Example 2.4.2

$$\mathfrak{R}: \begin{array}{lcl} (x * y) * z & \to & x * (y * z) \\ 1 * x & \to & x \\ i(x) * x & \to & 1 \\ x/y & \to & x * i(y) \end{array}$$

$$\begin{array}{lcl} [1]() & = & 2 \\ [*](x,y) & = & 2xy + x \\ [i](x) & = & x^2 \\ [/](x,y) & = & 2xy^2 + x + 1 \end{array}$$

∎

Example 2.4.3 Associativity and endomorphism

$$
\begin{array}{lll}
\mathfrak{R}: & f(f(x,y),z) & \to f(x,f(y,z)) \\
 & f(g(x),g(y)) & \to g(f(x,y)) \\
 & f(g(x),f(g(y),z)) & \to f(g(f(x,y)),z)
\end{array}
$$

$$
\begin{array}{lll}
[f](x,y) & = & xy + x \\
[g](x) & = & x + 1
\end{array}
$$

■

Example 2.4.4

$$
\begin{array}{lll}
\mathfrak{R}: & f(0) & \to 1 \\
 & f(s(x)) & \to g(f(x)) \\
 & g(x) & \to x + s(x)
\end{array}
$$

$$
\begin{array}{lll}
[0]() & = & 2 \\
[1]() & = & 2 \\
[f](x) & = & x^2 \\
[s](x) & = & 2x \\
[g](x) & = & 3x + 1 \\
[+](x,y) & = & x + y
\end{array}
$$

■

Example 2.4.5 ([Mu80])

$$
\begin{array}{lll}
\mathfrak{R}: & if(true,x,y) & \to x \\
 & if(false,x,y) & \to y \\
 & not(x) & \to if(x,false,true) \\
 & and(x,y) & \to if(x,y,false) \\
 & or(x,y) & \to if(x,true,y) \\
 & imply(x,y) & \to if(x,y,true) \\
 & equiv(x,y) & \to if(x,y,not(y)) \\
 & equiv(x,x) & \to true
\end{array}
$$

$$
\begin{array}{lll}
[true]() & = & 1 \\
[false]() & = & 1 \\
[if](x,y,z) & = & x + y + z \\
[not](x) & = & x + 3 \\
[and](x,y) & = & x + y + 2 \\
[or](x,y) & = & x + y + 2 \\
[imply](x,y) & = & x + y + 2 \\
[equiv](x,y) & = & x + 2y + 4
\end{array}
$$

■

Example 2.4.6

$\mathfrak{R}$:

$$\begin{array}{lll} \text{if}(\text{true}, x, y) & \rightarrow & x \\ \text{if}(\text{false}, x, y) & \rightarrow & y \\ \text{if}(x, y, y) & \rightarrow & y \\ \text{if}(\text{if}(x, y, z), u, v) & \rightarrow & \text{if}(x, \text{if}(y, u, v), \text{if}(z, u, v)) \end{array}$$

$$\begin{array}{lll} [\text{true}]() & = & 1 \\ [\text{false}]() & = & 1 \\ [\text{if}](x, y, z) & = & xy + xz + x \end{array}$$

■

Example 2.4.7 ([Ch89])

$\mathfrak{R}$:

$$\begin{array}{lll} f(f(x, y), z) & \rightarrow & f(x, f(y, z)) \\ f(e_j, y) & \rightarrow & y \\ f(x, i_j(x)) & \rightarrow & e_j \end{array} \quad \text{for all } j \in [1, n], n \geq 2$$

$$\begin{array}{llll} [f](x, y) & = & 2xy + x & \\ [e_j]() & = & 2 & j \in [1, n] \\ [i_1](x) & = & x^2 & \\ [i_j](x) & = & 5x^2 + 1 & j \in [2, n] \end{array}$$

■

Example 2.4.8 ([Ma87])

$\mathfrak{R}$:

$$\begin{array}{lll} (x * y) + (x * z) & \rightarrow & x * (y + z) \\ (x + y) + z & \rightarrow & x + (y + z) \\ (x * y) + ((x * z) + u) & \rightarrow & (x * (y + z)) + u \end{array}$$

$$\begin{array}{lll} [+](x, y) & = & xy + x \\ [*](x, y) & = & x + y \end{array}$$

Note that the system $\mathfrak{R}$ cannot be oriented in the desired way with the help of the recursive path ordering. ■

Example 2.4.9 Flattening

$\mathfrak{R}$:

$$\begin{array}{lll} \text{nil} \circ y & \rightarrow & y \\ (x.y) \circ z & \rightarrow & x.(y \circ z) \\ \text{flatten}(\text{nil}) & \rightarrow & \text{nil}.\text{nil} \\ \text{flatten}(x.y) & \rightarrow & \text{flatten}(x) \circ \text{flatten}(y) \\ \text{macflatten}(\text{nil}, y) & \rightarrow & \text{nil}.y \\ \text{macflatten}(x.y, z) & \rightarrow & \text{macflatten}(x, \text{macflatten}(y, z)) \end{array}$$

$$
\begin{aligned}
[nil]() &= 2 \\
[.](x,y) &= 2x + y + 1 \\
[\circ](x,y) &= 2x + y \\
[flatten](x) &= 4x \\
[macflatten](x,y) &= 3x + y
\end{aligned}
$$

■

Example 2.4.10 Dutch national flag ([De87])

$$
\begin{aligned}
\mathfrak{R}:\quad f(white, red) &\to f(red, white) \\
f(blue, red) &\to f(red, blue) \\
f(blue, white) &\to f(white, blue)
\end{aligned}
$$

$$
\begin{aligned}
[white]() &= 2 \\
[red]() &= 1 \\
[blue]() &= 3 \\
[f](x,y) &= 2x + y
\end{aligned}
$$

■

Example 2.4.11 ([Kü90])

$$
\begin{aligned}
\mathfrak{R}:\quad p(0) &\to 0 \\
p(s(x)) &\to x \\
s(x) > 0 &\to true \\
0 > y &\to false \\
s(x) > s(y) &\to x > y \\
0 = 0 &\to true \\
0 = s(y) &\to false \\
s(x) = 0 &\to false \\
s(x) = s(y) &\to x = y \\
if(true, x, y) &\to x \\
if(false, x, y) &\to y \\
s(x) > y &\to if(y = 0, true, x > p(y)) \\
x > s(y) &\to if(x = 0, false, p(x) > y)
\end{aligned}
$$

$$
\begin{aligned}
[p](x) &= x + 1 \\
[s](x) &= 2x + 4 \\
[>](x,y) &= xy + 1 \\
[=](x,y) &= x + y \\
[if](x,y,z) &= x + y + z \\
[0]() &= 1 \\
[true]() &= 1 \\
[false]() &= 1
\end{aligned}
$$

This rule system cannot be oriented with the help of any path ordering.

■

Example 2.4.12

$$
\mathfrak{R}: \quad
\begin{array}{lcl}
x * (y + z) & \rightarrow & (x * y) + (x * z) \\
(x + y) * z & \rightarrow & (x * z) + (y * z) \\
(x + y) + z & \rightarrow & x + (y + z) \\
0 + 1 & \rightarrow & 1 + 0 \\
1 + (0 + z) & \rightarrow & 0 + (1 + z) \\
1 + (2 + z) & \rightarrow & 2 + (1 + z) \\
1 + 2 & \rightarrow & 2 + (a + 1)
\end{array}
$$

Note that there is no polynomial ordering which can guarantee the termination of $\mathfrak{R}$ (cf. 6.4). ■

2.5 Discussion

In this chapter we have analyzed some properties of polynomial orderings. Motivated by the lack of a (general) formal definition for polynomial orderings we have attempted to unify the existing approaches by determining the necessary criteria for polynomial orderings to be reduction orderings. Consequently, the two traditional ways of defining polynomial orderings, by restricting the domain of the polynomials to $\mathbb{N}$ or requiring the subterm property, have turned out to be special cases of the one described here.

Besides this theoretical result we have shown that it is very easy to deduce a more restrictive, but, simultaneously feasible characterization from this definition. We have proposed an extended version of the polynomial orderings described in [Be86] and [BL87] that allows polynomials to range over $\mathbb{R}^{\geq 1}$. Instead of possessing the subterm property those polynomials have to be non-diminishing.

An interesting result is that, although we have a general characterization of all potential interpretations we have not found a way to extend the existing versions by, for example, allowing negative coefficients without simultaneously complicating the test for suitability of a polynomial. Thus, from a practical point of view we believe that lemma 2.2.10 should form the basis for future studies of polynomial orderings.

3 How to Choose the Interpretations

One of the main problems concerning polynomial orderings is to choose the right interpretation for a given rewrite system. It is very difficult to develop techniques for solving this problem. In this chapter we present some well-known heuristics of [Be86] and several new ones. Also, an algorithm for choosing appropriate linear polynomials is given. This method is based on the Simplex algorithm which will be explained in 3.4.

Note that all results presented in this chapter refer to polynomial interpretations that attach coefficients, taken from the set of natural numbers, to polynomials.

3.1 Some Well-known Heuristics

In this section, we give some suggestions for finding an appropriate interpretation which proves the termination of a given rewrite system. These kinds of heuristics are described in [Be86]. During experiments the main method used there was by "trial and error" and for this the help of the REVE-system was essential. The polynomial orderings are implemented as simplification orderings and their coefficients are natural numbers (which are greater than 0). In order to illustrate the suggestions we will apply them to an example (3.1.8) at the end of this section.

Suggestion 3.1.1

Consider the rules of the form

$$t \rightarrow t/u$$

where $u \in O(t) \setminus \{\varepsilon\}$.
Any interpretation of the function symbols guarantees the termination of such a rule since the polynomial ordering has the subterm property. ■

Suggestion 3.1.2

Assuming there is an operator which occurs in only one (a few, resp.) rule(s).
These kinds of function symbols cannot endanger the termination of the whole system. Therefore, a simple interpretation suffices. ■

Suggestion 3.1.3

The interpretation of each constant symbol will be 2:

$$(\forall c \in \mathfrak{F}) \ [c](\,) = 2.$$ ■

This proposal is based on a method for comparing two polynomials (see section 4.2). The assignment will improve the technique.

Suggestion 3.1.4

Orienting an associativity law like

$$f(f(x,y),z) = f(x,f(y,z))$$

there are two different interpretations of f:

- $f[f[x,y],z] \rightarrow f[x,f[y,z]]$
  if $[f](x,y) = axy + x$ or $[f](x,y) = bx^i + y$
  The basic idea is to give more "weight" to the first argument.

- $f[x,f[y,z]] \rightarrow f[f[x,y],z]$
  if $[f](x,y) = axy + y$ or $[f](x,y) = x + by^i$

such that a is an arbitrary integer and $b > 1 \lor i > 1$. ∎

It is noteworthy that the polynomials interpreting the operator f of the last suggestion could be generalized by adding any constant.

Suggestion 3.1.5

For all rules

$$f[t_1,\dots,t_n] \rightarrow t$$

defining an operator f where t does not contain any occurrence of f, let us use the following interpretation:

$$[f](x_1,\dots,x_n) = [t] + 1$$

Note that t could reduce the set of variables of $f[t_1,\dots,t_n]$. To preserve the monotony of f, we have to add the missing variables to $[f]$: $[f](x_1,\dots,x_n) = [t] + 1 + \sum_i x_i$. ∎

Suggestion 3.1.6

Consider the class of rules

$$f[g[t_1],t_2] \rightarrow g[f[t_1,t_2]].$$

To guarantee the termination of such rules use

$$[f](x,y) = xy \quad \text{and} \quad [g](x) = x + a$$

such that a is an arbitrary integer. Note that this suggestion depends on the requirement that the value of the interpretation of a variable or a constant symbol is at least 2 (see 3.1.3). ∎

Suggestion 3.1.7

The rule system contains a rule defining a homeomorphism:

$$g(f(t_1,...,t_n)) \rightarrow f(g(t_1),...,g(t_n)).$$

If we take

$$[f](x_1,...,x_n) = \sum_i x_i \quad \text{and} \quad [g](x) = x^{j+1}$$

(j is an arbitrary positive integer) the termination of such rules will be guaranteed. ∎

Example 3.1.8

In order to illustrate the power of the suggestions, we will apply them to the following pragmatic rule system:

$x + 0$	$\rightarrow$	x	(R1)
$0 + y$	$\rightarrow$	y	(R2)
$1 * y$	$\rightarrow$	y	(R3)
$x * 1$	$\rightarrow$	x	(R4)
$f(x) * f(y)$	$\rightarrow$	$f(x * y)$	(R5)
$(x * y) * z$	$\rightarrow$	$x * (y * z)$	(R6)
$-(x + y)$	$\rightarrow$	$(-x) + (-y)$	(R7)
$g(x * y, y)$	$\rightarrow$	$h(x * y, x)$	(R8)
$h(1, y)$	$\rightarrow$	y	(R9)

(R1) - (R4) and (R9) are terminating due to the suggestion 3.1.1 , (R5) due to 3.1.2, (R6) due to 3.1.4, (R8) due to 3.1.5 and (R7) due to 3.1.7. These considerations lead to the following interpretations:

$[0]()$	=	2
$[1]()$	=	2
$[f](x)$	=	$x + 1$
$[-](x)$	=	x^2
$[+](x, y)$	=	$x + y + 1$
$[*](x, y)$	=	$xy + x$
$[g](x, y)$	=	$2x + y + 1$
$[h](x, y)$	=	$x + y + 1$

∎

3.2 Several new Heuristics

We have integrated the polynomial orderings in our completion system COMTES (see [AMS89], [St89c], [Pa89], [AGGMS87]). A series of about 300 experiments has been conducted to gain more insight into the choice of interpretations (see [SK90]). A preliminary summary of the main results is given in the following table:

		%
Total number of rule systems tested	303	100
Canonical rule systems generated	225	74
Completion processes diverge (the termination can be guaranteed with a polynomial ordering)	28	9
Completion processes stop with failure, but the initial rule systems are orientable with a polynomial ordering	20	7
Rule systems which cannot usefully be oriented w.r.t. any polynomial ordering	30	10

The table indicates that the polynomial orderings provide a powerful technique for proving termination of term rewriting systems (since three quarters of the 303 examples are orientable). However, we cannot infer heuristics for generating adequate interpretations from these statistics. Therefore, we shall refine this diagram by splitting the examples into various classes and provide some assertions about the interpretations we will need to use. The next definition will classify the possible interpretations:

Definition 3.2.1

Let $p(x_1,\ldots,x_n)$ be any polynomial over $\mathbb{R}^n$, $a_i, c \in \mathbb{R}$.

- $p(x_1,\ldots,x_n)$ is linear

  iff $p(x_1,\ldots,x_n) = \sum_{i=1}^{n} a_i x_i + c$
  e.g.: $5x_1 + 3x_4 + x_5$

- $p(x_1,\ldots,x_n)$ is super-linear

  iff $p(x_1,\ldots,x_n) = \sum_{i=1}^{n} a_i x_i + c$ such that $a_i \in \{0,1\}$
  e.g.: $x_1 + x_3 + 1$

- $p(x_1,\dots,x_n)$ is separate

  iff each monomial of $p(x_1,\dots,x_n)$ consists of one variable, only
  e.g.: $2x_1^3 + 3x_2 + x_3^2$

- $p(x_1,\dots,x_n)$ is mixed

  iff $p(x_1,\dots,x_n)$ contains at least one monomial which is based on more than one variable
  e.g.: $x_1^2 + x_1x_3x_4^3 + 2$ ■

The following table lists different interpretations employed for the orientation of the examples. We made great efforts to develop minimal classes of interpretations, e.g. if a system can either be oriented with the help of a separate or a linear interpretation we will prefer the linear one.

		%
Total number of rule systems oriented	273	100
Rule systems oriented using super-linear interpretations	89	33
Rule systems oriented using linear but not super-linear interpretations	104	38
Rule systems oriented using separate but not linear interpretations	16	6
Rule systems oriented using mixed interpretations	64	23

The tested examples belong to several domains. The list of these classes will be given including typical representatives and appropriate interpretations.

- Algebraic structures: Groups, Rings, etc.

Example 3.2.2 Taussky group

$$
\begin{aligned}
x * (y * z) &\rightarrow (x * y) * z \\
1 * 1 &\rightarrow 1 \\
x * i(x) &\rightarrow 1 \\
g(x * y, y) &\rightarrow f(x * y, x) \\
f(1, y) &\rightarrow y
\end{aligned}
$$

$$
\begin{aligned}
[*](x,y) &= 2xy + y \\
[f](x,y) &= x + y \\
[g](x,y) &= 2xy^2 + y^2 + x + 1 \\
[i](x) &= x^2 \\
[1]() &= 2
\end{aligned}
$$

■

- Boolean theories: This class is closely connected with that of algebraic structures.

Example 3.2.3 Boolean ring

$$
\begin{aligned}
x \supset y &\rightarrow (x * y) + (x + 1) \\
x \vee y &\rightarrow (x * y) + (x + y) \\
x \equiv y &\rightarrow x + (y + 1) \\
\neg x &\rightarrow x + 1
\end{aligned}
$$

$$
\begin{aligned}
[+](x,y) &= x + y \\
[*](x,y) &= x + y \\
[\supset](x,y) &= 2x + y + 2 \\
[\vee](x,y) &= 2x + 2y + 1 \\
[\equiv](x,y) &= x + y + 2 \\
[\neg](x) &= x + 2 \\
[1]() &= 1
\end{aligned}
$$

■

Arithmetic theories: Addition, Multiplication, etc.

Example 3.2.4 Square number

$$
\begin{aligned}
x + 0 &\rightarrow x \\
x + s(y) &\rightarrow s(x + y) \\
double(0) &\rightarrow 0 \\
double(s(x)) &\rightarrow s(s(double(x))) \\
0^2 &\rightarrow 0 \\
(s(x))^2 &\rightarrow x^2 + s(double(x))
\end{aligned}
$$

$$
\begin{aligned}
[+](x,y) &= x + 2y \\
[s](x) &= x + 3 \\
[double](x) &= 3x \\
[^2](x) &= x^2 \\
[0]() &= 2
\end{aligned}
$$

■

- Lists: Append, Reverse, Flatten, etc.

Example 3.2.5 Iterative version of the reverse function ([HO80])

$$
\begin{array}{lll}
nil \circ y & \rightarrow & y \\
(x.y) \circ z & \rightarrow & x.(y \circ z) \\
rev(nil) & \rightarrow & nil \\
rev(x.y) & \rightarrow & rev(y) \circ (x.nil) \\
reviter(nil, y) & \rightarrow & y \\
reviter(x.y, z) & \rightarrow & reviter(y, x.z) \\
(x \circ y) \circ z & \rightarrow & x \circ (y \circ z) \\
rev(x) \circ y & \rightarrow & reviter(x, y) \\
rev(x) & \rightarrow & reviter(x, nil)
\end{array}
$$

$$
\begin{array}{lll}
[.](x, y) & = & xy + y + 1 \\
[\circ](x, y) & = & xy + x \\
[rev](x) & = & 2x^2 \\
[reviter](x, y) & = & xy + x \\
[nil]() & = & 2
\end{array}
$$

∎

- String rewriting systems: A Thue system $\mathfrak{T}$ over a set of strings Σ^* is a finite set of rules, each of the form $l \rightarrow r$, where l and r are words in Σ^*. Σ^* is the monoid freely generated by a finite alphabet Σ under the operation of concatenation. A close relation between term rewriting and Thue systems will exist if monadic terms are used only. A monadic term only contains unary function symbols and either a constant or a variable. The subset of the monadic terms without constants can unequivocally be transformed into strings and vice versa.

Example 3.2.6 Fibonacci group with five elements

$$
\begin{array}{lll}
c & \rightarrow & ab \\
d & \rightarrow & bc \\
e & \rightarrow & cd \\
de & \rightarrow & a \\
ea & \rightarrow & b
\end{array}
$$

$$
\begin{array}{lll}
[a](x) & = & x + 1 \\
[b](x) & = & 2x + 4 \\
[c](x) & = & 2x + 6 \\
[d](x) & = & 4x + 17 \\
[e](x) & = & 8x + 41
\end{array}
$$

∎

- Other systems: Systems that are not part of the five former classes.

Example 3.2.7

$$g(x, y) \rightarrow h(x, y)$$
$$h(f(x), y) \rightarrow f(g(x, y))$$

$$[f](x) = x + 2$$
$$[g](x, y) = 2x + y + 1$$
$$[h](x, y) = 2x + y$$

■

The next diagram presents a refinement of the statistics on the number of orientable examples based on the given classes.

Diagram 3.2.8 Evaluation of the tested examples

	(1)	(2)	(3)	(4)	(5)	(6)	(A)	(B)	(C)	(D)
Algebraic structures	60	23	70	8	17	5	12	26	4	58
Boolean theories	28	43	64	21	0	15	46	17	12	25
Arithmetic theories	69	65	73	4	0	23	21	51	13	15
Lists	36	72	81	8	8	3	20	46	3	31
String rewrite systems	30	53	83	7	3	0	43	57	0	0
Other systems	80	58	76	11	5	8	54	34	4	8

All numbers (except those of the first column) of the above diagram represent the percentages w.r.t. the first column. The encircled quantities have the following meanings:

(1) Total number of rule systems tested

(2) Initial rule systems are confluent

(3) Canonical rule systems generated

(4) Completion processes diverge (the termination can be guaranteed with a polynomial ordering)

(5) Completion processes stop with failure but the initial rule systems are orientable with a polynomial ordering

(6) Rule systems which cannot usefully be oriented w.r.t. any polynomial ordering

The following categories relate the different interpretations to the classes of orientable rule systems.

(A) Rule systems oriented using super-linear interpretations

(B) Rule systems oriented using linear but not super-linear interpretations

(C) Rule systems oriented using separate but not linear interpretations

(D) Rule systems oriented using mixed interpretations ∎

The most important results of this diagram can be summarized as follows:

- In contrast to the other classes, only a small quantity of initial systems in the class of algebraic structures is directly confluent. Moreover, the number of mixed interpretations needed is relatively big (nearly 60%).
- The proofs of the termination of the examples on lists and strings can often be successful conducted by using polynomial orderings. However, while lists require more than 30% of mixed interpretations, the string rewriting systems can be oriented with the help of linear interpretations, only (a possible explanation is that the string systems possess an easier structure).
- For the orientation of examples belonging to arithmetic theories mixed interpretations do not seem to be needed very often. Unfortunately, this does not agree with reality since only three quarters of the rule systems belonging to arithmetic theories are orientable.
- It is significant that three quarters of the general systems ('other systems') can be oriented. Furthermore, this orientation has been carried out using almost linear interpretations, only. It could be a lucky chance.

Based on the experience with the examples several new heuristics have been developed for the choice of useful interpretations.

Suggestion 3.2.9 Arithmetic theories

In order to prove the termination of arithmetic specifications it is profitable if the complexity hierarchy of the interpretations reflects the complexity hierarchy of the corresponding function symbols (cf. 3.2.4). ∎

For example, $[+](x,y) = xy + x$, $[*](x,y) = x^2y + xy^2$, $[exp](x,y) = (x + y)^4$ is based on suggestion 3.2.9.

The following suggestion provides a test for deciding whether a mixed interpretation is necessary for orientation.

Suggestion 3.2.10 Mixed interpretations

Let $l = f(s_1,\ldots,s_m) \rightarrow g(t_1,\ldots,t_n) = r$ be a rule such that $m > 1, n \geq 1$. There exists no separate interpretation $[f]$ such that $l \geq_{POL} r$ if

$$
\begin{aligned}
(\exists s_i) \quad & V(s_i) \cap \bigcup_{j=1,j\neq i}^{m} V(s_j) = \emptyset \\
& \wedge\ (\exists u \in O(r))\ \ r/u = f(s_1',\ldots,s_m') \ \wedge\ s_i' \sqsupseteq s_i \\
& \wedge\ V(r[u.i \leftarrow \lambda]) \cap V(s_i) \neq \emptyset
\end{aligned}
$$
∎

The first condition requires that the variables of s_i do not occur elsewhere in l. The last requirement means that at least one variable of $V(s_i)$ occurs in r outside of s_i'. We now demonstrate the use of this test with the help of an example.

Example 3.2.11 Binomial coefficients

The following rule system represents a complete specification of the binomial coefficients:

$$
\begin{aligned}
b(0,s(y)) &\rightarrow 0 \\
b(x,0) &\rightarrow s(0) \\
b(s(x),s(y)) &\rightarrow b(x,s(y)) + b(x,y)
\end{aligned}
$$

There exists no separate interpretation of b which orients the last rule. Let

$$
\begin{aligned}
l &= b(s(x),s(y)) \quad , \quad r = b(x,s(y)) + b(x,y) \\
s_i &= s(y) \\
r/u &= b(x,s(y))
\end{aligned}
$$

Note that all conditions of 3.2.10 are valid: y does not occur elsewhere in l,

$s_i = s(y) \sqsubset s(y) = s_i'$ and $V(b(x,\lambda)+b(x,y)) \cap V(s(y)) = \{y\}$. Therefore, we need a mixed interpretation of b to prove the termination of the system:

$$[b](x,y) = xy + x$$
$$[+](x,y) = x + y$$
$$[s](x) = 2x$$
$$[0]() = 2$$

■

Approximately 20% of the 273 systems orientable w.r.t. polynomial orderings require mixed interpretations. The criterion above can be applied to nearly 50% of these cases as, for example, to the distributivity axiom.

During experimentation the combination of the distributivity and the associativity proved to be problematic. The following suggestion can sometimes decide the termination of a system containing these two rules.

Suggestion 3.2.12 Distributivity axioms

Let $*, + \in \mathfrak{F}$, c be a constant symbol and

$$\mathfrak{R}_D: \quad x * (y + z) \to (x * y) + (x * z)$$
$$(x + y) * z \to (x * z) + (y * z)$$

- The termination of $\mathfrak{R}_D$ can be proved with the help of

$$[+](x,y) = x + y + 1$$
$$[*](x,y) = xy$$
$$[c]() \geq 2$$

- The system

$$\mathfrak{R}_D \cup \{x + (y + z) \to (x + y) + z\}$$

terminates if

$$[+](x,y) = x + 2y + 1$$
$$[*](x,y) = xy$$
$$[c]() \geq 2$$

- The rule system

$$\mathfrak{R}_D \cup \{(x * y) * z \to x * (y * z)\}$$

can be oriented using the following interpretations:

$$\begin{aligned}[+](x,y) &= x + y + 2\\ [*](x,y) &= xy + x\\ [c]() &\geq 3\end{aligned}$$

- In order to achieve the termination of

$$\mathfrak{R}_D \cup \{(x + y) + z \rightarrow x + (y + z)\\ (x * y) * z \rightarrow x * (y * z)\}$$

we use the interpretations

$$\begin{aligned}[+](x,y) &= 2x + y + 4\\ [*](x,y) &= xy + x\\ [c]() &\geq 5\end{aligned}$$

■

Finally, we deal with a large and wide-spread area: group theory. As mentioned previously (see 3.2.8), this class mainly requires mixed interpretations which are considerably harder to obtain.

Suggestion 3.2.13 Group theory

Let $*, i, e \in \mathfrak{F}$. Furthermore, let E be a set of group axioms including

$$\begin{aligned}x * (y * z) &= (x * y) * z \quad \text{and}\\ i(x * y) &= i(y) * i(x).\end{aligned}$$

An appropriate and relatively simple interpretation for orienting E is the following one:

$$\begin{aligned}[*](x,y) &= 2xy + x \quad (\text{or } 2xy + y)\\ [i](x) &= x^2\\ [e]() &= 2\end{aligned}$$

Note that no system specifying a group can be oriented using separate interpretations, only. ■

3.3 A Method for Linear Polynomials

In the last two sections we have presented several suggestions for choosing the interpretations of function symbols. The disadvantage of these techniques is that

- they are very vague and incomplete and
- it is hard to combine them,

i.e. an automatic method would be inefficient since it must be able to backtrack.

To partially overcome these problems we have developed a procedure that mechanically generates interpretations of operators such that a given rule system terminates. This technique is restricted to special polynomials, i.e. the following procedure can only handle linear interpretations. This restriction is derived from the fact that it is very difficult to check whether a general polynomial is greater than 0 (cf. chapter 4). Since 71% of the interpretations used for the examples are linear (see section 3.2) we consider it to be a valid restriction.

Example 3.3.1

Let $s = f(g(g(x)), y)$ and $t = f(g(x), f(x, y))$ be two terms. Furthermore, let

$$[f](x, y) = ax + by + c \quad \text{and} \quad [g](x) = dx + e$$

be two classes of linear interpretations for f and g, respectively. Then,

$$[s] = ad^2x + by + ade + ae + c\ ,$$
$$[t] = (ad + ab)x + b^2y + ae + bc + c\ .$$

Thus, proving $s >_{POL} t$ requires the proof of $[s] >_P [t]$. ∎

Lemma 3.3.2

Let $a = \sum_{i=1}^{n} a_i x_i + a_0$ and $b = \sum_{i=1}^{n} b_i x_i + b_0$ be linear polynomials with $(\forall i \in [1,n])\ a_i, b_i \geq 0$. Then,

$$a \gtrsim_P b$$
$$\text{if} \quad (\forall i \in [0, n]) \quad a_i \geq b_i$$

∎

Example 3.3.3 (Example 3.3.1 continued)

$$
\begin{array}{lll}
[s] \gtrsim_P [t] & & \\
\text{if} & & ad^2 \geq ad + ab \\
& \wedge & b \geq b^2 \\
& \wedge & ade + ae + c \;\geq\; ae + bc + c
\end{array}
$$

Note that we have to compare sums of products. There is a relatively simple algorithm, the so-called Simplex method (see 3.4), for deciding whether a system of linear inequalities has a solution. Unfortunately, we do not have linear inequalities, in general. Consequently, we shall present a method for transforming such an expression (a sum of products) into a linear polynomial. ■

Given an inequality of the form used in lemma 3.3.2, its transformation to a linear inequality is based on the following ideas:

- approximating each side of the inequality to a product and then
- applying a logarithmic function to these products.

The following lemmata provide the theoretical framework for the procedure.

Lemma 3.3.4

$$\prod_{i=1}^{n} a_i \geq \sum_{i=1}^{n} a_i$$

$$\text{if} \quad (\forall i \in [1,n]) \quad a_i \geq 2$$

■

Lemma 3.3.5

$$n \cdot \prod_{i=1}^{n} a_i \geq \sum_{i=1}^{n} a_i$$

$$\text{if} \quad (\forall i \in [1,n]) \quad a_i \geq 1$$

■

Lemma 3.3.6

$$\frac{1}{n} \cdot \sum_{i=1}^{n} a_i \geq \left(\prod_{i=1}^{n} a_i \right)^{\frac{1}{n}}$$

$$\text{if} \quad (\forall i \in [1,n]) \quad a_i \geq 0$$

This inequality is called the arithmetic-mean-geometric-mean inequality for n numbers. ■

The combination of these three lemmata leads to the method of transforming an inequality (of lemma 3.3.2) into an inequality of only two products:

Lemma 3.3.7

Let $a_i \geq 1$ and $b_i \geq 1$. Then,

$$\sum_{i=1}^{n} a_i \geq \sum_{i=1}^{m} b_i$$

if i) $n^n \cdot \prod_{i=1}^{n} a_i \geq \prod_{i=1}^{m} b_i^n \;\wedge\; (\forall i \in [1,m]) \;\; b_i \geq 2$

or ii) $n^n \cdot \prod_{i=1}^{n} a_i \geq m^n \cdot \prod_{i=1}^{m} b_i^n$ ∎

Example 3.3.8 (Example 3.3.3 continued)

We have to verify that $ad^2 \geq ad + ab$, $b \geq b^2$ and $ade + ae + c \geq ae + bc + c$:

i) $ad^2 \geq ad + ab$ will be transformed into
$ad^2 \geq a^2bd \;\wedge\; ad \geq 2 \;\wedge\; ab \geq 2$
or $ad^2 \geq 2a^2bd$

ii) $b \geq b^2$ will be transformed into
$b \geq b^2 \;\wedge\; b^2 \geq 2$
or $b \geq b^2$

iii) $ade + ae + c \geq ae + bc + c$ will be transformed into
$27a^2cde^2 \geq a^3b^3c^6e^3 \;\wedge\; ae \geq 2 \;\wedge\; bc \geq 2 \;\wedge\; c \geq 2$
or $27a^2cde^2 \geq 27a^3b^3c^6e^3$ ∎

Remarks 3.3.9

- The inequality of the first transformation (i) of the example above could be weakened by dividing it by its greatest common factor:

  $ad^2 \geq ad + ab \;\leadsto\; d^2 \geq d + b \;\leadsto\; d^2 \geq bd \;\wedge\; \ldots$
  or $d^2 \geq 2bd$

- If $n = m = 1$, the first possibility (i) of lemma 3.3.7 of transforming an inequality is more restricted than the second one (cf. ii) of example 3.3.8):

  i) $a_1 \geq b_1 \;\wedge\; b_1 \geq 2$
  or ii) $a_1 \geq b_1$

  Therefore, if there is only one product on both sides of the inequality, we do not need any transformation.

- The inequality of the third transformation (iii) of the example above could be weakened by subtracting the greatest common part:

$$ade + ae + c \geq ae + bc + c \quad \rightsquigarrow \quad ade \geq bc$$ ∎

Notation 3.3.10

The transformation according to lemma 3.3.7 of an inequality to an inequality where both sides contain only one product will be denoted by 'red': $red(\Sigma a_i \geq \Sigma b_i)$. ∎

In order to use the Simplex method, we have to map $red(\Sigma a_i \geq \Sigma b_i)$ into a linear inequality. For this purpose we exploit a special property of logarithmic functions:

$$\log(\Pi a_i) = \Sigma \log(a_i).$$

Furthermore, log is monotonous if $a_i \geq 1$. Note that, for guaranteeing the monotony of the polynomial orderings on linear interpretations, all coefficients must be greater than or equal to 1. Therefore, the condition $a_i \geq 1$ is always fulfilled.

For eliminating exponents we utilize the law $\log(u^n) = n \cdot \log(u)$. According to lemma 3.3.7 it is convenient to apply the binary logarithm *lb* since inequalities of the form $a_i \geq 2$ can then be transformed into $lb(a_i) \geq 1$.

Lemma 3.3.11

$$\prod_{i=1}^{n} a_i \geq \prod_{i=1}^{m} b_i$$

$$\text{iff} \quad \sum_{i=1}^{n} lb(a_i) \geq \sum_{i=1}^{m} lb(b_i)$$ ∎

Notation 3.3.12

The transformation due to lemma 3.3.11 of an inequality of two products into a linear inequality is denoted by 'lin': $lin(\Pi a_i \geq \Pi b_i)$. ∎

With the help of the transformation functions *red* and *lin* we are able to reduce an inequality (relative to linear interpretations) to a linear inequality. Therefore, the test whether a term s is greater than a term t (w.r.t. polynomial orderings) produces a set of linear inequalities. For deciding whether such a set has a solution the Simplex method can be applied. If the set of inequalities has a solution and we are not only interested in a solution, we have to reverse the effect of the logarithmic transformation by applying lin^{-1} to the result:

$$\begin{aligned} lin^{-1}(\Sigma\ lb(a_i) \geq \Sigma\ lb(b_i)) &= (2^{\Sigma\ lb(a_i)} \geq 2^{\Sigma\ lb(b_i)}) \\ &= (\Pi\ 2^{lb(a_i)} \geq \Pi\ 2^{lb(b_i)}) \\ &= (\Pi\ a_i \geq \Pi\ b_i)\ . \end{aligned}$$

Algorithm 3.3.13

We present an algorithm to determine whether a finite set of rewrite rules $\mathfrak{R} = \{l_i \to r_i \mid i \in [1,n]\}$ can be ordered by a polynomial ordering on linear interpretations on function symbols.
The algorithm is presented by using inference rules of the form

$$\frac{A}{B} \quad \text{cond}$$

which means that B is valid if A and cond are true.

(1) Transforming the rules ($\mathfrak{R}$) into inequalities (I)

$$\frac{\mathfrak{R} \cup \{l \to r\}\ ,\ I}{\mathfrak{R}\ ,\ I \cup \{[l] > [r]\}}$$

This step will be executed until $\mathfrak{R}$ is empty, i.e. I contains all inequalities. This transformation must combine the coefficients of common variables.

(2) Splitting the inequalities w.r.t. the coefficients of common variables (cf. lemma 3.3.2)

$$\frac{I \cup \{\sum_{i=1}^{n} a_i x_i + a_0 > \sum_{i=1}^{n} b_i x_i + b_0\}}{I \cup \{a_i \geq b_i \mid i \in \{0,\dots,j-1,j+1,\dots,n\}\} \cup \{a_j > b_j\}}$$

(3) Simplifying the inequalities (cf. remarks 3.3.9)

$$\frac{I \cup \{x + \Sigma\, a_i \geq x + \Sigma\, b_i\}}{I \cup \{\Sigma\, a_i \geq \Sigma\, b_i\}}$$

- $$\frac{I \cup \{\Sigma x \cdot a_i \geq \Sigma x \cdot b_i\}}{I \cup \{\Sigma a_i \geq \Sigma b_i\}}$$

- $$\frac{I \cup \{a \geq 0\}}{I}$$

(4) Eliminating the addition (cf. lemma 3.3.7 and notation 3.3.10)

$$\frac{I \cup \{a \geq b\}}{I \cup \{red(a \geq b)\}}$$ if a or b contain +

(5) Reducing a product to a linear polynomial (cf. lemma 3.3.11 and notation 3.3.12)

$$\frac{I \cup \{a \geq b\}}{I \cup \{lin(a \geq b)\}}$$ if a and b do not contain +

(6) Renaming the logarithm functions

$$\frac{I \cup \{\Sigma a_i\, lb(x_i) \geq \Sigma b_i\, lb(y_i)\}}{I \cup \{\Sigma a_i x_i' \geq \Sigma b_i\, y_i'\}}$$ if a_i and b_i are constants

(7) Applying the Simplex method

$$\frac{I}{I'}$$ if I' is the output of the application of the Simplex algorithm to the set I of linear inequalities

(8) Reversing the logarithm mapping

$$\frac{I \cup \{\Sigma a_i x_i' \geq \Sigma b_i y_i'\}}{I \cup \{\Pi\, 2^{a_i} \cdot x_i \geq \Pi\, 2^{b_i} \cdot y_i\}}$$ if a_i, b_i are constants and I contains the solution (provided the Simplex method has found one) ■

In order to improve the efficiency of this algorithm we apply the criterion of section 3.2 (cf. 3.2.10) to the rules of the rule system. That way, systems which require mixed interpretations can be excluded straight away.

Examples 3.3.14

i) We would like to guarantee the termination of the rule (example 3.3.1)

$$f(g(g(x)), y) \rightarrow f(g(x), f(x, y)):$$

Let $[f](x, y) = ax + by + c$ and $[g](x) = dx + e$.

(1) $I = \{ad^2x + by + ade + ae + c > (ad + ab)x + b^2y + ae + bc + c\}$

(2) $I = \{ad^2 \geq ad + ab \, , \, b \geq b^2 \, , \, ade + ae + c \geq ae + bc + c\}$

(3) $I = \{d^2 \geq d + b \, , \, 1 \geq b \, , \, ade \geq bc\}$

(4) $I = \{d^2 \geq bd \, , \, b \geq 2 \, , \, d \geq 2 \, , \, 1 \geq b \, , \, ade \geq bc\}$

Producing this set of inequalities we tried the first alternative of lemma 3.3.7. But this set does not have any solution since the subset $\{b \geq 2 \, , \, 1 \geq b\}$ is unsolvable. Thus, we use the second transformation of lemma 3.3.7:

$I = \{d^2 \geq 2bd \, , \, 1 \geq b \, , \, ade \geq bc\}$

(5) $I = \{2lb(d) \geq 1 + lb(b) + lb(d) \, , \, 0 \geq lb(b) \, ,$
$lb(a) + lb(d) + lb(e) \geq lb(b) + lb(c)\}$

Note that if there is any inequality like $x \leq 1$, then the value of x has to be set to 1 since the precondition requires $x \geq 1$, for all x. Furthermore, substitute all occurrences of x by 1.

(6) $I = \{2d' \geq 1 + b' + d' \, , \, 0 \geq b' \, , \, a' + d' + e' \geq b' + c'\}$

(7) A possible solution of this set of linear inequalities is the following one:

$$b' = 0 \, , \, d' = 1 \, , \, a' = c' = e' = 0$$

Note that the last inequality $(a' + d' + e' \geq b' + c')$ is the proper inequality which is needed (cf. (2) of algorithm 3.3.13).

(8) $b = 2^0 = 1 \, , \, d = 2^1 = 2 \, , \, a = c = e = 2^0 = 1$

These assignments generate the interpretations for f and g in the following way: $[f](x, y) = x + y + 1$, $[g](x) = 2x + 1$. With the help of these interpretations, $f(g(g(x)), y) >_{POL} f(g(x), f(x, y))$ since $[f(g(g(x)), y)] = 4x + y + 4 > 3x + y + 3 = [f(g(x), f(x, y))]$.

ii) We would like to prove the termination of the Fibonacci function:

$$\begin{aligned} fib(0) &\rightarrow 0 \\ fib(s(0)) &\rightarrow s(0) \\ fib(s(s(x))) &\rightarrow fib(s(x)) + fib(x) \end{aligned}$$

Let be $[0]() = a$, $[s](x) = bx + c$, $[fib](x) = dx + e$ and $[+](x, y) = px + qy + r$.

(1) $I = \{ad + e > a,\ abd + cd + e > ab + c,\ b^2dx + bcd + cd + e > (bdp + dq)x + cdp + ep + eq + r\}$

(2) $I = \{ad + e > a,\ abd + cd + e > ab + c,\ b^2d \geq bdp + dq,\ bcd + cd + e \geq cdp + ep + eq + r\}$

(3) $I = \{ad + e > a,\ abd + cd + e > ab + c,\ b^2 \geq bp + q,\ bcd + cd + e \geq cdp + ep + eq + r\}$

(4) $I = \{4ade > a^2,\ 27abcd^2e > 8a^3b^3c^3,\ b^2 \geq 2bpq,\ 27bc^2d^2e \geq 64c^3d^3e^6p^6q^3r^3\}$

(5) $I = \{2 + lb(a) + lb(d) + lb(e) > 2lb(a),\ lb(27) + lb(a) + lb(b) + lb(c) + 2lb(d) + lb(e) > 3 + 3lb(a) + 3lb(b) + 3lb(c),\ 2lb(b) \geq 1 + lb(b) + lb(p) + lb(q),\ lb(27) + lb(b) + 2lb(c) + 2lb(d) + lb(e) \geq 6 + 3lb(c) + 3lb(d) + 6lb(e) + 6lb(p) + 3lb(q) + 3lb(r)\}$

(6) $I = \{2 + a' + d' + e' > 2a',\ lb(27) + a' + b' + c' + 2d' + e' > 3 + 3a' + 3b' + 3c',\ 2b' \geq 1 + b' + p' + q',\ lb(27) + b' + 2c' + 2d' + e' \geq 6 + 3c' + 3d' + 6e' + 6p' + 3q' + 3r'\}$

(7) The set of inequalities of (6) is unsolvable. Therefore, this method cannot prove the termination of the Fibonacci function. ▪

Theorem 3.3.15

The algorithm 3.3.13 always terminates. If it does not fail $\mathfrak{R}$ can be ordered by a polynomial ordering (using only linear polynomials). ▪

The power of the presented method strongly depends on transformation (4), i.e. on lemma 3.3.7. This lemma approximates each side of an inequality to a product such that the following relations hold: Find $a = \Pi\, c_i$ and $b = \Pi\, d_i$ with

$$\Sigma a_i \geq a \geq b \geq \Sigma b_i.$$

Accordingly, the better the approximations are the more powerful the algorithm will be.

Note that it is possible to simplify the inequalities generated by step (4) with the rules of step (3). Therefore, we can merge step (3) and step (4).

Another improvement is the extension of transformation (3) which is used to simplify an inequality. The following rules can in some cases avoid the algorithm to stop unsuccessfully:

- $$\frac{I \cup \{\prod_{i=1}^{n} a_i \le 1\}}{I \cup \{a_i = 1 \mid i \in [1,n]\}}$$

- $$\frac{I \cup \{a_i = 1\}}{I' \cup \{a_i = 1\}}$$ if I' results from I by substituting a_i by 1

- $$\frac{I \cup \{\Sigma\, a_i \ge \Sigma\, b_i\}}{I}$$ if $(\exists \pi_1, \pi_2)(\forall i)(\exists c)\ a_{\pi_1(i)} = b_{\pi_2(i)} \cdot c$

By way of illustration, the last rule removes for example $a_1c_1 + a_2c_2 \ge a_1 + a_2$ from the set of inequalities since it always holds.

The three transformations above extend the presented method. Furthermore, there is another simplification rule which sometimes improves the technique:

$$\frac{I}{I \cup \{a_0 = 0\}}$$ if $(\exists f \in \mathfrak{F}, n > 0)\ [f](x_1,\dots,x_n) = \Sigma\, a_i x_i + a_0$

With the help of this condition, the transformation red of lemma 3.3.7 can be described more precisely. Consider for example the Fibonacci function of 3.3.14 ii):

(2) $I = \{ad + e > a\ ,\ abd + cd + e > ab + c\ ,\ b^2d \ge bdp + dq,$
$bcd + cd + e \ge cdp + ep + eq + r\}$

(3) $I = \{d > 1\ ,\ d > 1\ ,\ b^2 \ge bp + q\ ,\ c = 0\ ,\ e = 0\ ,\ r = 0\}$

(4) $I = \{d > 1\ ,\ d > 1\ ,\ b^2 \ge 2bpq\}$

(5) $I = \{lb(d) > 0\ ,\ lb(d) > 0\ ,\ 2lb(b) \ge 1 + lb(b) + lb(p) + lb(q)\}$

[6] I = {$d' > 0$, $d' > 0$, $2b' \geq 1 + b' + p' + q'$}

[7] A feasible solution of I is the following one:

$$b' = 1 \ , \ d' = 1 \ , \ p' = 0 \ , \ q' = 0$$

[8] $a = 1$, $b = 2$, $c = 0$, $d = 2$, $e = 0$, $p = 1$, $q = 1$ and $r = 0$ which lead to the interpretations $[0]() = 1$, $[s](x) = 2x$, $[fib](x) = 2x$, $[+](x,y) = x + y$ which prove the termination of the three rules describing the Fibonacci function.

3.4 The Simplex Algorithm

In 3.3 an algorithm has been presented which computes appropriate linear polynomials for guaranteeing the termination of rule systems for which linear interpretations suffice. An important part of this algorithm consists of solving linear inequalities. In the present section we give an informal description of a special version of the well-known Simplex method to solve this problem.

In 1947, Dantzig first proposed the Simplex method whereby a linear form could be minimized (or maximized) subject to linear equalities (or inequalities) that are sometimes called constraints. Such problems have come to be known as 'Linear Programming'. The following approach is taken from [Th79], [Ch83] and [Mi76] and the reader is referred to these references for a more detailed description.

The Simplex method can handle the following problem:

$$\text{Minimize} \quad \sum_{i=1}^{n} c_i x_i$$
$$\text{Subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad (i = 1,\dots,m)$$
$$x_j \ge 0 \qquad (j = 1,\dots,n)$$

For simplicity of exposition we shall restrict ourselves to the form above. It is not difficult to transform a more general form (in which equations of the form $\Sigma a_{ij}x_j = b_i$ as well as inequalities like $\Sigma a_{ij}x_j \ge b_i$ are possible) into the one used here.

To transform this problem into an equivalent form in which the inequalities are replaced by equalities, m so-called slack variables $x_{n+1},\dots,x_{n+m}$ are introduced as distinct from the n so-called decision variables in which the problem is defined:

$$\text{Minimize} \quad z = \sum_{i=1}^{n} c_i x_i$$
$$\text{Subject to} \quad \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i \qquad (i = 1,\dots,m)$$
$$x_j \ge 0 \qquad (j = 1,\dots,n+m)$$

In a linear programming problem, the linear function z to be optimized is called the objective function. Any point $(x_1,\dots,x_n)$ with non-negative coordinates that satisfies the system of constraints is called a feasible solution to the problem. Thus, our basic problem is to determine, from among the set of all feasible solutions, a point that minimizes the objective function. The Simplex method can decide whether a problem has, in fact, any feasible solution and in addition whether the objective function actually assumes a minimum value. Note, however, that the problem appearing in the algorithm of section 3.3 consists of finding *any* solution of a system of linear equalities, i.e. we shall only study

$$[*] \qquad \sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i \qquad (i = 1,\dots,m)$$

Solving such systems is no more difficult than solving linear programming problems: to find a solution of [*], or to establish its non-existence, we only need

Minimize	$z = x_0$	
Subject to	$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0 = b_i$	$(i = 1,\dots,m)$
	$x_0 \geq 0$	
		Fig. 1

The basic step of the Simplex method is derived from the familiar pivot operation used to solve linear equations. The pivot operation consists of replacing a system of equations with an equivalent system in which a selected variable is eliminated from all but one of the equations.

Definition 3.4.1 Pivoting

Let $\sum_{j=1}^{n} a_{kj}x_j + x_{n+k} - x_0 = b_k$ be the k-th equality of figure 1. We choose any x_p $[p \in [1,n]]$ and rewrite it in terms of x_{n+k}, i.e.

$$x_p = (b_k - \sum_{j=1, j\neq p}^{n} a_{kj}x_j - x_{n+k} + x_0)/a_{kp}$$

Substituting x_p in the other equations a new equation set is obtained. This operation represents a change of state and will be denoted by pivot(p,k). ■

It is easy to show that the solution set of the system of equations resulting from the pivot operation is identical to the solution set of the original system. In general, repeated use of pivoting can lead to a system of equations whose solution set is obvious. Such a system (called canonical form) consists of n equations with n unknowns where each variable appears in one and only one equation, and in that equation has coefficient one. However, in attempting to put the constraint system into canonical form, an arbitrary selection of decision variables could easily lead to a system with some negative constant terms and thus to an associated solution that is not even feasible. Therefore, for solving the problem of 3.3, it is not sufficient to use only pivot operations (like in Gaussian elimination) in some way. The Simplex method cleverly applies a convenient pivot operation at the right time.

What must be developed is a technique for determining an initial feasible solution for an arbitrary system of equations. The basic idea behind the method used to solve this problem is simple. We introduce a sufficient number of variables, called artificial variables, to put the system of constraints into canonical form with these variables as the decision variables. Then, we apply the Simplex method to a new objective function defined in such a way that its minimum value corresponds to a feasible solution of the original problem.

Definition 3.4.2 Introducing artificial variables

The transformation of the system (see figure 1)

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0 &= b_i \qquad (i = 1,\ldots,m) \\ x_0 &= z \end{aligned}$$

into the system containing the artificial variables $x_{n+m+1},\ldots,x_{n+2m}$

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0 + x_{n+m+i} &= b_i \qquad (i = 1,\ldots,m) \\ x_0 &= z \\ -\sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0\Big) &= w - \sum_{i=1}^{m} b_i \end{aligned}$$

Fig. 2

will be denoted by *canonical transformation*. ∎

Note that the system of constraints of figure 2 is canonical with the artificial variables as decision variables. The new objective function $w = x_{n+m+1} + \ldots + x_{n+2m}$ is transformed into canonical form by subtracting each equation of the system of constraints from $w = x_{n+m+1} + \ldots + x_{n+2m}$.

If the pivot operations dictated by the problem of minimizing w are simultaneously performed on the equation z which defines the original objective function, this function will be expressed in each step in terms of variables which are no decision variables. Thus, if an initial basic feasible solution is found for w, the Simplex method can be initiated immediately on z.

We now present the steps of the Simplex algorithm, starting with a problem in canonical form:

$$
\begin{array}{lllllllll}
\text{Minimize} & & & c_{m+1}x_{m+1} & + \dots + & c_n x_n & = & z \\
\text{Subject to} & x_1 + & & a_{1m+1}x_{m+1} & + \dots + & a_{1n}x_n & = & b_1 \\
& & x_2 + & a_{2m+1}x_{m+1} & + \dots + & a_{2n}x_n & = & b_2 \\
& & \ddots & & & & & \\
& & x_m + & a_{mm+1}x_{m+1} & + \dots + & a_{mn}x_n & = & b_m
\end{array}
$$

Algorithm 3.4.3 Simplex algorithm

stop $\leftarrow$ **false**
repeat
  if $(\forall j \in [m+1, n])\ \ c_j \geq 0$
  then stop $\leftarrow$ **true** (success)
  else **if** $(\exists j \in [m+1, n])(\forall i \in [1, m])\ \ c_j < 0 \ \wedge\ a_{ij} \leq 0$
    then stop $\leftarrow$ **true** (failure)
    else pivot(p, k)
      with p $\leftarrow$ column with the smallest negative c_j
        k such that $b_k/a_{kp} = \min\{b_i/a_{ip} \mid a_{ip} > 0\}$
until stop ■

Note that p could be any column with a negative c_j term. The smallest c_j can reduce the total number of steps necessary to complete the problem. Furthermore, if the minimum of b_i/a_{ip} is attained in several rows, a simple rule (such as choosing that row with the smallest index) can be used to determine the pivoting row.

The Simplex method presented in 3.4.3 is correct and terminates. There are some specific complications while applying this procedure. For compactness we would like to refer to literature for a detailed description of these problems.

Until now we have presented the various devices needed for solving the problem of finding a solution of a system of linear equations. Before applying these methods to an example we will construct an algorithm from them that can solve our problem of section 3.3.

Algorithm 3.4.4

Input: $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ $\qquad (i = 1,\dots,m)$

$\Downarrow$ Introducing slack variables (figure 1)

$$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0 = b_i \qquad (i = 1,\dots,m)$$

$\Downarrow$ Canonical transformation (Definition 3.4.2)

$$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0 + x_{n+m+i} = b_i \qquad (i = 1,\dots,m)$$

$$\sum_{i=1}^{m} b_i - \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} - x_0\right) = w$$

$\Downarrow$ Applying the Simplex method for generating a basic feasible solution (Algorithm 3.4.3)

Output: Values for $x_1,\dots,x_n$ that solve the input problem
or Failure ∎

Note that in algorithm 3.4.4, we reject the original objective function $z = x_0$. This function's only use is to justify the employment of the Simplex method for solving systems of linear inequalities, only. It is irrelevant for producing a basic solution of our problem.

The following example is intended to demonstrate the essential steps of the algorithm above. It is a slight modification of an example contained in [Th79].

Example 3.4.5

We are interested in a solution for the following system of linear inequalities:

$$\begin{array}{l} x_1 - 2x_2 - 3x_3 - 2x_4 \geq 3 \\ x_1 - x_2 + 2x_3 + x_4 \leq 11 \end{array}$$

$\Downarrow$ Adding slack variables

$$\begin{array}{l} -x_0 + x_1 - 2x_2 - 3x_3 - 2x_4 - x_5 = 3 \\ -x_0 + x_1 - x_2 + 2x_3 + x_4 + x_6 = 11 \end{array}$$

$\Downarrow$ Introducing artificial variables

$$\begin{array}{l} -x_0 + x_1 - 2x_2 - 3x_3 - 2x_4 - x_5 + x_7 = 3 \\ -x_0 + x_1 - x_2 + 2x_3 + x_4 + x_6 + x_8 = 11 \\ \hline x_7 + x_8 = w \end{array}$$

Canonical transformation

$$\begin{array}{l} -x_0 + \mathbf{x_1} - 2x_2 - 3x_3 - 2x_4 - x_5 + x_7 = 3 \\ -x_0 + x_1 - x_2 + 2x_3 + x_4 + x_6 + x_8 = 11 \\ \hline 2x_0 - 2x_1 + 3x_2 + x_3 + x_4 + x_5 - x_6 = w - 14 \end{array}$$

Simplex method: $x_1 = 3 + x_0 + 2x_2 + 3x_3 + 2x_4 + x_5 - x_7$

$$\begin{array}{l} -x_0 + x_1 - 2x_2 - 3x_3 - 2x_4 - x_5 + x_7 = 3 \\ x_2 + \mathbf{5x_3} + 3x_4 + x_5 + x_6 - x_7 + x_8 = 8 \\ \hline - x_2 - 5x_3 - 3x_4 - x_5 - x_6 + 2x_7 = w - 8 \end{array}$$

Simplex method: $x_3 = \frac{8}{5} - \frac{1}{5}x_2 - \frac{3}{5}x_4 - \frac{1}{5}x_5 - \frac{1}{5}x_6 + \frac{1}{5}x_7 - \frac{1}{5}x_8$

$$\begin{array}{l} -x_0 + x_1 - \frac{7}{5}x_2 - \frac{1}{5}x_4 - \frac{2}{5}x_5 + \frac{3}{5}x_6 + \frac{2}{5}x_7 + \frac{3}{5}x_8 = \frac{39}{5} \\ \frac{1}{5}x_2 + x_3 + \frac{3}{5}x_4 + \frac{1}{5}x_5 + \frac{1}{5}x_6 - \frac{1}{5}x_7 + \frac{1}{5}x_8 = \frac{8}{5} \\ \hline x_7 + x_8 = w \end{array}$$

Now, let the slack variables as well as the artificial variables be of value zero. This implies the following equalities:

$$\begin{aligned} x_1 - \frac{7}{5}x_2 \qquad - \frac{1}{5}x_4 &= \frac{39}{5} \\ \frac{1}{5}x_2 + x_3 + \frac{3}{5}x_4 &= \frac{8}{5} \end{aligned}$$

The easiest solution is $x_1 = \frac{39}{5}$, $x_2 = 0$, $x_3 = \frac{8}{5}$ and $x_4 = 0$. Another one consists of the vector (19, 8, 0, 0). ∎

At each step of the Simplex method it is sufficient to know only the coefficients of the variables in the system of equations. In particular, for computation by hand or simple computer implementations it is favourable to record this information, only. A representation known as *Contracted Tableau* or *Tucker-diagram* is of the following form:

x_1	x_2	$\dots$	x_n	
a_{11}	a_{12}	$\dots$	a_{1n}	b_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
a_{m1}	a_{m2}	$\dots$	a_{mn}	b_m
c_1	c_2	$\dots$	c_n	c

Tableau

The first m rows correspond to the system of constraints with the constant terms given in the last column. The last row corresponds to the equation defining the objective function with the constant term (on the right-hand side of that equation) in the last column. The z term of the objective function is suppressed from the tableau as it remains fixed throughout the Simplex method.

There are two other forms of tableau representations known as *Extended tableau* and *Tucker-Beale form*. The reader is referred to [Mi76] for a formal description of these diagrams and to other publications in the field of linear programming for more details about the Simplex algorithm.

3.5 Discussion

This chapter dealt with strategies for the generation of appropriate interpretations which guarantee the termination of a given rule system.

First of all, we presented some heuristics by Ben Cherifa ([Be86]) including suggestions for orienting associative laws, rules that define operators ($f(t_1,...,t_n) \rightarrow t$ such that t does not contain f) and homeomorphism rules.

Section 3.2 contains some new suggestions about arithmetic theories, distributivity axioms and group theories. Furthermore, we presented a criterion by which the necessity of mixed interpretations can sometimes be detected. All these insights have been gained from an analysis of about 300 examples. Detailed statistics on these examples can also be found in this section.

A partial improvement of the heuristics presented in the sections 3.1 and 3.2 is contained in 3.3. It is a procedure that mechanically generates interpretations of operators. This technique is restricted to linear interpretations (note that nearly three quarters of the interpretations used to orient the 300 examples have a linear form). The basic ideas of the algorithm are the following ones:

- transforming the rules into inequalities by using interpretations with variables as coefficients
- approximating each side of the inequalities to a product
- applying a logarithm function to these products (note that we now have linear inequalities)
- using the Simplex method (see section 3.4) for solving these linear inequalities

Note that the algorithm of section 3.3 cannot directly be transformed to general interpretations since it is very difficult to generalize lemma 3.3.2. We believe that it is possible to extend this technique to separate interpretations. Furthermore, this method can be improved by incorporating the heuristics of sections 3.1 and 3.2.

4 How to Check the Positiveness of Polynomials

The use of polynomial orderings reduces the proof of termination of rewrite systems to appropriate interpretations orienting the given system on the one hand and to whether a given polynomial is greater than another one on the other hand.

In the last chapter we discussed heuristics and an algorithm for finding appropriate polynomials based on a given set of rules.

The basis of this chapter is the presentation of two well-known procedures ([BL87], [Ro88]) for deciding the positiveness of polynomials. This is equivalent to decide whether one of two given polynomials is greater than the other.

4.1 The Problem

The proof of a given polynomial in n variables being positive over real numbers is generally decidable even though in exponential time ([Ta51], [Co75]). However, if we restrict the domain of the polynomial to a proper subset of ℝ, such as ℕ, the problem is generally undecidable ([Da73]). Nevertheless, since we are interested in whether a polynomial has a root in $[\Gamma_G]$ it is to be expected that a decision procedure, to our problem, will not be found.

Instead of trying to solve inequalities it would be more convenient to search for appropriate algorithms that check the properties which insure the wanted inequalities. This can be done in various ways, as will be shown.

Let us first look at $[\Gamma_G]$. Since we are primarily concerned with theorem 2.3.4, $[\Gamma_G]$ is a proper subset of $\mathbb{R}^{\geq 1}$. Our first simplification will be to test the positiveness of a polynomial on $\mathbb{R}^{\geq 1}$ instead of $[\Gamma_G]$. It should be noted that this requirement is sufficient albeit unnecessary.

In the following pages we propose two different algorithms (see [BL87], [Ro88]) to perform this task. Both of these techniques originally presume $[\mathfrak{F}]$ to be a subset of $\mathfrak{Pol}(\mathbb{N})$, the set of polynomials having coefficients of ℕ, although they can also be used for proving the positiveness of polynomials over $\mathbb{R}^{\geq 1}$.

The computations associated with the first method of Ben Cherifa and Lescanne utilize an elementary and basic principle to devise a simple and efficient implementation ([BL87]).

The second more powerful method of Rouyer utilizes a more analytical procedure which is based on the theorem of Sturm ([Ro88], [Du60]).

4.2 Comparing Polynomials by Comparing Monomials

The following method is that proposed by Ben Cherifa and Lescanne ([BL87]). The main idea is to prove $p > 0$ by finding polynomials $p_0, p_1, \ldots, p_n$ such that $p = p_0 \geq p_1 \geq \ldots \geq p_n > 0$. The positiveness of p_n is supposed to be checked by a basic principle like 'all coefficients are positive'.

To illustrate how this method works let us have a look at the following example.

Example 4.2.1

Suppose we have to prove that a term s is greater than a term t w.r.t. a polynomial interpretation [...] such that

$$[s] = x^2y^2 + 4y^2 + 2xy^2 \quad \text{and}$$
$$[t] = x^2y^2 + y^2 + xy + 2y + x.$$

If we want to orient the equation $s = t$ to $s \rightarrow t$, we need to check the positiveness of the polynomial

$$\begin{aligned} p = [s] - [t] &= x^2y^2 + 4y^2 + 2xy^2 - x^2y^2 - y^2 - xy - 2y - x \\ &= 3y^2 + 2xy^2 - xy - 2y - x \end{aligned}$$

This can be done by finding for each monomial m_1 of p having a negative coefficient, a monomial m_2 whose value is greater than or equal to the absolute value of m_1 for all instantiations of x and y. If m_2 has this property ($m_2 \geq |m_1|$ for all instantiations of x and y) it is said to bound m_1. Clearly, a monomial m_1 can be bounded only by a monomial m_2 containing each variable occurring in m_1 in at least the same power as in m_1.
Since we consider polynomials over $\mathbb{R}^{\geq 1}$, $3y^2$ bounds $-2y$. Now there are $-xy$ and $-x$ left to be bounded by $2xy^2$. If we split $2xy^2$ into $xy^2 + xy^2$ we see that the first xy^2 bounds $-xy$ and the second one bounds $-x$.

In summary, we have

$$\begin{aligned} p = [s] - [t] &= x^2y^2 + 4y^2 + 2xy^2 - x^2y^2 - y^2 - xy - 2y - x \\ &= 3y^2 + 2xy^2 - xy - 2y - x \\ &> 2xy^2 - xy - x && \text{since } 3y^2 > |-2y| \\ &= xy^2 + xy^2 - xy - x \\ &\geq xy^2 - x && \text{since } xy^2 \geq |-xy| \\ &\geq 0 && \text{since } xy^2 \geq |-x| \end{aligned}$$

Note that the strict inequality $p > 0$ only holds since $3y^2 > |-2y|$. If we had $[s] = x^2y^2 + 3y^2 + 2xy^2$ and, therefore, $p = 2y^2 + 2xy^2 - xy - 2y - x$

we could not prove the positiveness of p in the same manner since for $x = y = 1$ we have $[s] = 6 = [t]$. ∎

This last remark provides a convenient starting point for the presentation of the algorithm.

Whereas in the second chapter we gave a definition of polynomial orderings by restricting the set of interpretations as little as possible, we now wish to work in the opposite direction by requiring any constant c to be mapped to a real number $C \geq 2$ by every polynomial interpretation [...]. This implies, we exclude interpretations [...] with $[c] < 2$ for any constant c.

The reason for this apparent requirement is readily understood. Since all polynomials are non-diminishing, it follows that $[t] \geq 2$ for each $t \in \Gamma$ and thus, for each monomial $m = c_{r_1 \ldots r_n} X_1^{r_1} \ldots X_n^{r_n}$ it holds that $c_{r_1 \ldots r_n} X_1^{r_1} \ldots X_i^{r_i} \ldots X_n^{r_n} \geq 2 \cdot c_{r_1 \ldots r_n} X_1^{r_1} \ldots X_i^{r_i - 1} \ldots X_n^{r_n}$. For example, $x^2 \geq 2x > x$ and $x^2y \geq 2xy > xy$.

It should be noted that, by increasing the lower bound for the interpretation of constants, this method becomes more powerful, although this needs to be postponed.

There are two main problems encountered. Firstly, we have to choose a pair of monomials, one having a positive and the other one containing a negative coefficient. Secondly, we must set the negative one against the positive one. Since these two procedures, named CHOOSE and CHANGE, are the essential points of the problem, we will discuss them in detail after the presentation of the whole algorithm.

We assume that the polynomial can be represented as a set of monomials each realized as a tuple $(c_{e1 \ldots en}, e_1, \ldots, e_n)$ where the e_i's stand for the exponents of the variables x_i and $c_{e1 \ldots en}$ for the coefficient of the monomial.

Algorithm 4.2.2 ([BL87])

```
POSITIVE = proc (P : polynomial) returns (string)
    while there exists a negative coefficient do
          if   there exist c_e1...en > 0  and  c_f1...fn < 0
               with ei ≥ fi  for all  i ∈ {1,...,n}
          then CHOOSE(c_e1...en , c_f1...fn)
               CHANGE(c_e1...en , c_f1...fn)
          else return ("no answer")
    end
return ("positive")
end
```
∎

As noted in the example the main idea of the procedure POSITIVE is to consider a monomial with a negative coefficient, say m, and to try to find a monomial with a positive coefficient, say m', which bounds it. When such a monomial m' is found we divide it into two parts m'_1 and m'_2 with $m'_1 + m'_2 \leq m'$ such that m'_1 bounds m.

Thus, to prove the positiveness of the whole polynomial p we can replace the monomials m and m' by m'_2 getting p' and prove the positiveness of p'. For example, we prove 2xy - xy - x to be positive by transforming 2xy to xy + xy, replacing 2xy and -xy by xy and proving xy - x to be positive.

We now propose the function CHANGE. As mentioned previously the realization of CHANGE depends heavily on the lower bound for the interpretation of the constants. In the following, we take for granted that the constants will be interpreted by real numbers greater than or equal to 2 [cf. suggestion 3.1.3].

Thus, a monomial m having $c_{e1...en}$ as coefficient bounds a monomial m' consisting of the coefficient $c_{f1...fn}$ exactly if $c_{e1...en} \cdot 2^{e1-f1} \cdot ... \cdot 2^{en-fn} > |c_{f1...fn}|$. If m does not bound m', this number can be added to $c_{f1...fn}$ to minimize the negative coefficient. Since we can describe each monomial by its coefficient CHANGE could be defined as follows:

```
CHANGE = proc(c_e1...en, c_f1...fn : monomial)
    if   c_e1...en > |c_f1...fn · 2^(f1-e1) · ... · 2^(fn-en)|
    then c_e1...en := c_e1...en + c_f1...fn · 2^(f1-e1 + ... + fn-en)
         c_f1...fn := 0
    else c_f1...fn := c_f1...fn + c_e1...en · 2^(e1-f1 + ... + en-fn)
         c_e1...en := 0
end
```

Let us now consider the function CHOOSE. Given a polynomial consisting of a set of positive and a set of negative monomials, CHOOSE realizes a heuristic for finding an appropriate positive monomial for each negative monomial. This means that, for example, to attach the positive monomial x^2 instead of $2x^2y$ to the negative monomial -x of $p = 2x^2y + x^2 + 1 - x - 2y$. If we use $2x^2y$ we cannot prove the positiveness of p. Whereas, choosing x^2 leads to $2x^2y + 1 - 2y$ that can readily be proved to be greater than 1. This choice establishes the positiveness of p.

This example shows that the success of the algorithm heavily depends on the realization of CHOOSE. Since in [BL87] the only remark on the implementation of CHOOSE is not particularly useful, we will discuss an extended version proposed in [Pa89].

CHOOSE first searches for a negative monomial that can be bounded by just one of the positive monomials. If so, these two monomials are taken. Otherwise, CHOOSE searches for a positive monomial that covers one negative monomial. If both conditions do not hold any pair of a negative and a positive monomial will be taken if the positive one contains each variable occurring in the negative one in, at least, the same power.

CHOOSE = **proc**($c_{e1...en}$, $c_{f1...fn}$: monomial)
  if (there exists $c_{g1...gn} < 0$ and $c_{h1...hn}$ is the only positive monomial with hi ≥ gi for all i ∈ {1,...,n})
    or
    (there exists $c_{h1...hn} > 0$ and $c_{g1...gn}$ is the only negative monomial with hi ≥ gi for all i ∈ {1,...,n})
    or
    (there exists $c_{h1...hn} > 0$ and $c_{g1...gn} < 0$ with hi ≥ gi for all i ∈ {1,...,n})
  then $c_{e1...en} := c_{h1...hn}$
    $c_{f1...fn} := c_{g1...gn}$
end

Example 4.2.3 ([Pa89])

Let $p = x^3yz + uz + 4x^2u + z - x^2 - xu - 2u - xz$.
CHOOSE will return the pairs $(-xu, 4x^2u)$ since no other positive monomial can bound $-xu$.

$$\rightsquigarrow \quad p_1 = x^3yz + uz + \frac{7}{2}x^2u + z - x^2 - 2u - xz$$

For the same reason we get the pair $(-xz, x^3yz)$

$$\rightsquigarrow \quad p_2 = \frac{7}{8}x^3yz + uz + \frac{7}{2}x^2u + z - x^2 - 2u$$

uz cannot cover any negative monomial except $-2u$

$$\rightsquigarrow \quad p_3 = \frac{7}{8}x^3yz + \frac{7}{2}x^2u + z - x^2$$

Neither the first nor the second condition holds, so the first possible monomial $(\frac{7}{8}x^3yz)$ is chosen to bound $-x^2$

$$\rightsquigarrow \quad p_n = \frac{3}{4}x^3yz + \frac{7}{2}x^2u + z$$ ■

The two first conditions checked in CHOOSE guarantee that the most appropriate pair of monomials is chosen if one of the conditions holds. This means that, if the algorithm fails because of a 'wrong' choice this wrong choice must be made at another point. Experiments showed that in practice nearly every time one of the two first conditions hold.

Finally, we propose that it should be quite simple to extend this algorithm in such a way that it automatically computes the lower bound of $[\Gamma_G]$ w.r.t. a given interpretation and uses this value instead of 2 in the procedure CHANGE. This would enable a user to measure the influence of this value on the results of the algorithm.

4.3 Comparing Polynomials by Comparing Signs

The basic concept of this method of Rouyer (see [Ro88]) for proving a polynomial to be positive over a real interval is the fact that a polynomial p(x), w.r.t. one variable x, has the root x_0 with multiplicity one if, and only if, it changes its sign at x_0. This fact, for example, can be used to determine whether a linear polynomial equals zero in the interval [a, b]. This would be true if the sign of p(a) is not equal to that of p(b).

Since

$$(\forall \text{ k-fold root } x_0 \text{ of } p(x)) \qquad p(x_0) = p'(x_0) = \ldots = p^{k-1}(x_0) = 0$$

(for which the exponent k describes the k-fold derivative of p) this idea can be extended and applied to the case of higher degree polynomials.

Starting with the polynomial to be analyzed one can build a so-called Sturm sequence of polynomials $(p_0, p_1, \ldots, p_n)$ by

$$\begin{aligned} p_0 &= p(x) \\ p_1 &= p'(x) \\ p_{i+2} &= -(p_i \bmod p_{i+1}) \qquad i \geq 0 \end{aligned}$$

where $p_i \bmod p_{i+1}$ represents the rest when dividing p_i by p_{i+1}. In the following theorem the idea mentioned above is formalized.

Theorem 4.3.1

Let a, b ∈ ℝ with a < b.
$N_p(a)$ ($N_p(b)$, resp.) denotes the number of sign changes in the Sturm sequence $p_0(x), \ldots, p_n(x)$ for x = a (x = b, resp.).

Then, the number of real roots in [a, b] is $N_p(a) - N_p(b)$. ∎

If, for example, $p_i(a) > 0$ for all i ∈ {1,...,n} and $p_i(b) < 0$ for all i ∈ {1,...,n}, then the number of real roots in [a, b] will be 0 - 0 = 0.

Since the calculation of a Sturm sequence in the above manner is costly (because of the Euclidian division) it is a significant step to explore a procedure for calculating Sturm sequences starting with p and p' that does not need any division (see [Du60]). The polynomials p_i computed by this method only differ to those computed using the first method by a positive factor and therefore could be used in connection with theorem 4.3.1.

In the following we describe the application of this method to our problem of proving a polynomial p[x] to be positive on a real interval [see [Ro88]].

Let $p_0[x] = p[x] = \sum_{i=0}^{n} a_i x^{n-i}$ and $p_1[x] = p'[x] = \sum_{i=0}^{n-1} b_i x^{n-i-1}$.

Firstly, we will record the coefficients of p_0 and p_1 in a table:

p_0	a_0	a_1	...	a_{n-1}	a_n
p_1	b_0	b_1	...	b_{n-1}	

Then, we calculate a third and a fourth line in the following manner:

$$[\forall i \in [0,n-1]] \quad c_i := b_0 a_{i+1} - a_0 b_{i+1}$$
$$[\forall i \in [0,n-2]] \quad d_i := c_0 b_{i+1} - b_0 c_{i+1}$$

The d_i's are the coefficients of the polynomial p_2 we searched for. Using p_1 and p_2, as p_0 and p_1 before, we again compute a line of c_i's and a line of d_i's which will represent p_3. This procedure will continue until one line consists only of zeros.

Example 4.3.2 ([Ro88])

Let $p_0[x] = x^3 - 2x^2 + x + 1$
$p_1[x] = 3x^2 - 4x + 1$

1	p_0	1	-2	1	1
2	p_1	3	-4	1	
3		-2	2	3	
4	p_2	2	-11		
5		25	2		
6	p_3	-279			

We get $p_2[x] = 2x - 11$
$p_3[x] = -279$

Note that for computing line five we use line two and line four. ∎

To get the number of real roots of p[x] that are greater than one, we will record the signs of $p_i[1]$ and $p_i[+\infty]$ for i = 0,1,2,3:

	1	∞
0	+	+
1	0	+
2	-	+
3	-	-
N_p	1	1

Since $N_p(1) - N_p(+\infty) = 1 - 1 = 0$ and $p(\infty) > 0$ it is proved that $p(x) > 0$ for $x \geq 1$.

Until now we were engaged in proving the positiveness of a polynomial in one variable. Is it possible to transfer this procedure to polynomials in n variables ? We will proceed in a way similar to that when defining partial derivatives of functions.

Given a polynomial $p(x_1,\dots,x_n)$ in n variables we treat $x_2,\dots,x_n$ as constants and get a polynomial $p(x_1)$ in one variable. For example, let $p(x, y) = x^2 + y^2 - 2xy + 1$. Then, $p(x,y) = p(x) = 1\cdot x^2 + (-2y)\cdot x + (y^2 + 1)$ such that 1, $-2y$, $y^2 + 1$ are the coefficients of $p(x)$. Thus, proving $p(x, y)$ to be positive is reduced to prove $p(x)$ to be positive.

Now we have the fact that $p(x) = \Sigma\, a_i x^{n-i} > 0$ for all $x \in [1,\infty[$ if, and only if, $p(\infty) > 0$ and $p(x)$ does not equal to zero in $[1,\infty[$. The first of these two conditions can readily be checked. The second one is an application of what we have introduced before. Let us consider the following example.

Example 4.3.3 ([Ro88])

Let $p(x, y) = x^2 + y^2 - 2xy + 1$.
Then, $p(x, y) = p(x) = 1\cdot x^2 + (-2y)\cdot x + (y^2 + 1)$

Proving $p(x)$ to be positive over $[1,\infty[$ requires $p(\infty)$ to be positive and $p(x)$ to be not equal to zero in $[1,\infty[$.
Since $p(\infty) > 0$, it is left to show that $p(x)$ is not zero in $[1,\infty[$. This could be done by building a Sturm sequence as described previously:

p_0	1	$-2y$	$y^2 + 1$
p_1	2	$-2y$	
	$-2y$	$2y^2 + 2$	
p_2	-4		

This diagram implies the sign variation table

	1	+∞
0	$sg(y^2 + 2 - 2y)$	+
1	$sg(2 - 2y)$	+
2	-	-
N_p	?	1

To get an answer to the question for the sign of p(1) we have to apply again our method to $y^2 + 2 - 2y$ and to $2 - 2y$. Clearly, we are hoping to find $sg(y^2 + 2 - 2y) = +$ or $sg(2 - 2y) = -$ (otherwise, the technique is unsuccessful). In that case $N_p(1)$ would be 1 and, therefore, we would have proved p(x,y) to be positive.

(a) $q_0(y) = y^2 + 2 - 2y$
$q_1(y) = 2y - 2$

q_0	1	-2	2
q_1	2	-2	
	-2	4	
	-4		

	1	∞
0	+	+
1	0	+
2	-	-
N_q	1	1

Thus, $y^2 + 2 - 2y$ has no sign change in $[1, \infty[$. Since $q_0(\infty) > 0$ we have $sg(y^2 + 2 - 2y) = +$.

(b) $r_0(y) = -2y + 2$
$r_1(y) = -2$

r_0	-2	2
r_1	-2	
	-4	
r_2	0	

	1	∞
0	-	-
1	-	-
2	0	0
N_r	0	0

Since $-2y + 2$ has no sign change in $[1, \infty[$ and $r_0(\infty) < 0$ we have $sg(-2y + 2) = -$.

We could now complete the above sign variation table for $p(x)$:

	1	∞
0	+	+
1	-	+
2	-	-
N_p	1	1

It is $N_p(1) - N_p(\infty) = 0$ and therefore $p(x)$ is not zero in $[1, \infty[$. In addition, $p(\infty) > 0$ which causes the positiveness of $p(x, y)$. ∎

Similar to the method proposed in 4.2 it is possible to extend this method by attaching a dummy value instead of real values to the constants. This means that the algorithm could test if the given rule system could be oriented by increasing the lower bound of $[\Gamma_G]$.

Probably this could lead to interesting information about the dependence of the comparison of polynomials on the values attached to the constants while leaving the remaining interpretation unchanged.

4.4 Discussion

In this chapter we have described two well-known procedures ([BL87], [Ro88]) for proving the positiveness of a polynomial. They both originally presume [$\mathfrak{F}$] to be a subset of $\mathfrak{Pol}(\mathbb{N})$ but they can also be used for proving the positiveness of polynomials over $\mathbb{R}^{\geq 1}$.

The first technique is based on the idea of finding a polynomial that is smaller than the given one but simultaneously positive ([BL87]).

The second one makes use of several fundamental analytical theorems ([Ro88], [Du60]). It appears to be more powerful but since we have not implemented this method in our COMTES-system, we cannot yet verify this opinion by experiments.

Both techniques have in common a remarkable dependance on the smallest value interpreting a constant. And both techniques will gain more power by increasing the lower bound of [$\Gamma_{\mathfrak{C}}$]. We think that an analysis of this dependency and especially an investigation as to whether a given polynomial ordering could be strengthened by manipulating (increasing) the coefficients could lead to more powerful methods for proving the positiveness of polynomials as well as new heuristics for finding appropriate interpretations.

5 Polynomial Orderings Modulo Theories

The use of term rewriting systems based on an additional underlying theory E presumes a special termination property. Here, we consider various theories and define appropriate polynomial orderings that can be used to prove E-termination. Some of these orderings are well-known ([Be86]), but we also present some new ones.

5.1 E-Termination

The basic concept of term rewriting systems is to convert an equation into a directed rewrite rule by comparing both sides w.r.t. an ordering. However, there exist equations which are incomparable in any case. For example, a rewriting system containing the commutativity axiom $x+y = y+x$ as a rule is non-terminating. If the termination property is not satisfied, the set of axioms can be split into two parts: The axioms causing non-termination are used as equations E while the others are used as rewrite rules $\mathfrak{R}$. An appropriate reduction relation allows reductions modulo the equations in E. The effective computation with this relation presumes

- a complete unification algorithm for the equational theory E and
- the E-termination, i.e. there is no infinite sequence of terms of the form $t_1 =_E t_1' \Longrightarrow_{\mathfrak{R}} t_2 =_E t_2' \Longrightarrow_{\mathfrak{R}} \dots$.

Several methods extending the classical Knuth-Bendix completion procedure have been developed. Some of them are described in [PS81], [Jo83] and [JK86].

Since this paper mainly deals with termination we adapt the general results on termination to the case of equational term rewriting systems (see, for example, [BP85], [De87], [Hu80b], [JM84], [LB77a], [LB77b] and [LB77c]).

An equational term rewriting system terminates if there is an ordering > which contains the rewrite relation $\Longrightarrow_{\mathfrak{R}/E} = =_E \cdot \Longrightarrow_{\mathfrak{R}} \cdot =_E$. The test of this inclusion requires all derivations of the form $s \Longrightarrow_{\mathfrak{R}/E} t$ to be checked. This requirement can be refined:

Lemma 5.1.1 ([BP85])

If > is E-compatible, then

$>$ contains $\Longrightarrow_{\mathfrak{R}/E}$ iff $>$ contains $\overset{+}{\Longrightarrow}_{\mathfrak{R}}$ ∎

Definition 5.1.2

An ordering > is E-compatible

iff

$$\begin{array}{ccc} s =_E s' & & s' \\ > & \text{implies} & > \\ t =_E t' & & t' \end{array}$$ ∎

If a reduction ordering $>$ is E-compatible and $\sigma(l) > \sigma(r)$ for every rule $l \rightarrow_{\mathfrak{R}} r$ and every substitution σ, then the equational term rewriting system $\mathfrak{R}/E$ terminates (see [BP85]).

Obviously, E-termination strongly depends on the given underlying theory E. For example, E must satisfy the following two conditions in order to prevent infinite derivations (see [JM84]):

- If $s = t \in E$, then the set of variables of both terms must be identical. Otherwise, there will be loops since we may instantiate the additional variable by an instance of a left-hand side of a rule of $\mathfrak{R}$. By rewriting the term with the rule and by then applying the 'starting equation' twice, the term will be derived from itself.
  For example,
  $\mathfrak{R}$: $x * 1 \rightarrow x$
  E: $x * 0 = 0$
  $0 =_E (x * 1) * 0 \Rightarrow_{\mathfrak{R}} x * 0 =_E 0$
- Furthermore, E-termination cannot be satisfied if there is an equation of the form $t =_E x$ such that x has more than one occurrence in t. In this case a left-hand side l of a rule of $\mathfrak{R}$ is E-equal to a term with several occurrences of l. Therefore, we can rewrite one of these and start the process again.
  For example,
  $\mathfrak{R}$: $\neg\neg x \rightarrow x$
  E: $x \wedge x = x$
  $\neg\neg x =_E \neg\neg x \wedge \neg\neg x \Rightarrow_{\mathfrak{R}} x \wedge \neg\neg x =_E x \wedge (\neg\neg x \wedge \neg\neg x) \Rightarrow_{\mathfrak{R}}$
  $x \wedge (x \wedge \neg\neg x) =_E \ldots$

Hence, the use of an ordering $>$ for rewriting systems modulo a theory E presumes the E-compatibility of $>$. There exist only a few orderings which have this property. Most of them are only compatible w.r.t. associative and commutative (AC, for short) theories.

For example, associative path orderings (see [Gn88], [GL86], [BP85] and [DHJP83]) extend the recursive path orderings to AC-congruence classes. They are based on flattening and transforming the terms by a rewriting system with rules similar to the distributivity axioms. Furthermore, the precedence on the operators has to satisfy a property called associative pair condition. A disadvantage of the associative path ordering is its inefficiency which results from the demand that two terms must be pre-processed (flattened and transformed w.r.t. the distributivity axioms) before they are compared. [St89d] and [St90b] contain an application of an extension (the embedding of status) of the associative path orderings to several path and decomposition orderings. Since these orderings are stronger than the recursive path ordering, the corresponding orderings restricted to AC-theories are more powerful than the associative path ordering.

A new class of orderings compatible with AC has been introduced in [St89b] and [St90a]. It is based on the Knuth-Bendix ordering with status (KBOS, see [St89a]). A modification of this ordering (called associative-commutative Knuth-Bendix ordering, ACK) causes its AC-commutation (which is weaker than the AC-compatibility), a property introduced by Jouannaud and Munoz ([JM84]). The most important aspects of this ordering are i) Multiset status is assigned to each commutative function symbol, ii) The weight of each associative operator which has to be minimal w.r.t. the precedence is zero and iii) The terms only have to be partly flattened to be compared. A major advantage of this technique is the possibility of applying the algorithm of [Ma87] to find a useful weight function for proving the termination of a given rewrite system. The power of the ordering is nearly the same as that of the Knuth-Bendix ordering with status.

5.2 Proving E-Termination Using Polynomial Orderings

It is well-known that polynomial orderings can also be restricted to AC-theories (see [Be86], [BL87], [BL86]). In order to guarantee the AC-compatibility the interpretations of AC-operators must be of a special form. The following lemma generalizes this concept:

Lemma 5.2.1

$>_{POL}$ is E-compatible if
$(\forall s,t \in \Gamma)\ \ s =_E t \ \ \rightsquigarrow\ \ s =_{POL} t$ ∎

This property allows the handling of many different theories, apart from AC. Ben Cherifa ([Be86]) tests a few theories for their compatibility. This chapter deals with the extension of her catalogue (including all results of [Be86]). In the rest of this chapter we present the conditions for the interpretations of E-operators under which the induced polynomial ordering is E-compatible. For simplicity, we use abbreviations for the theories:

Definition 5.2.2

Let $f, g, 1 \in \mathfrak{F}$.

· Associativity	:	$A(f)$	iff	$f(x,f(y,z))$	=	$f(f(x,y),z)$
· Commutativity	:	$C(f)$	iff	$f(x,y)$	=	$f(y,x)$
· Left Commutativity	:	$C_L(f)$	iff	$f(f(x,y),z)$	=	$f(f(x,z),y)$
· Right Commutativity	:	$C_R(f)$	iff	$f(x,f(y,z))$	=	$f(y,f(x,z))$
· Left Distributivity	:	$D_L(f,g)$	iff	$f(g(x,y),z)$	=	$g(f(x,z),f(y,z))$
· Right Distributivity	:	$D_R(f,g)$	iff	$f(x,g(y,z))$	=	$g(f(x,y),f(x,z))$
· Endomorphism	:	$E(f,g)$	iff	$f(g(x,y))$	=	$g(f(x),f(y))$
· Idempotency	:	$I(f)$	iff	$f(x,x)$	=	x
· Minus	:	$M(f)$	iff	$f(f(x))$	=	x
· Permutativity	:	$P(f)$	iff	$f(x_1,\dots,x_n)$	=	$f(x_{\pi(1)},\dots,x_{\pi(n)})$
· Transitivity	:	$T(f,g)$	iff	$f(g(x,y),g(y,z))$	=	$f(g(x,y),g(x,z))$
· Left Unipotency	:	$U_L(f,1)$	iff	$f(1,x)$	=	x
· Right Unipotency	:	$U_R(f,1)$	iff	$f(x,1)$	=	x ∎

These theories were adopted from [Si89]. Combinations of these theories are given in 5.12. Furthermore, assertions about special theories like special groups, rings, etc. are contained in 5.13.

5.3 Associativity

Lemma 5.3.1 ([BL87])

$$A(f) \quad \text{if} \quad [f](x,y) = a_1xy + a_2(x+y) + a_3 \quad \text{and} \quad a_1a_3 + a_2 = a_2^2$$ ∎

Special cases:

- $[f](x,y) = axy$ (if $a_2 = a_3 = 0$) for arbitrary values of a
- $[f](x,y) = x + y + c$ (if $a_1 = 0 \wedge a_2 = 1$) for arbitrary values of c

Example 5.3.2 ([KS83])

$\mathfrak{R}$:

$$
\begin{aligned}
(x/x)/((y/y)/y) &\rightarrow y\\
(x/y)/(z/y) &\rightarrow x/z\\
x/x &\rightarrow 1\\
1/x &\rightarrow i(x)\\
x/i(y) &\rightarrow x * y
\end{aligned}
$$

E: $(x * y) * z = x * (y * z)$

I:

$$
\begin{aligned}
[1]() &= 1\\
[i](x) &= x^2\\
[/](x,y) &= 2xy^2 + x + y^2 + 1\\
[*](x,y) &= 2xy + x + y
\end{aligned}
$$ ∎

5.4 Commutativity

There are three kinds of commutativity axioms, the (classical) commutativity, the left and the right commutativity.

Lemma 5.4.1

$$C(f) \quad \text{if} \quad [f](x,y) = \sum_{i,j} a_{ij}\, x^i y^j \;\wedge\; (\forall i,j)\; a_{ij} = a_{ji}$$ ∎

Special cases:

- $[f](x,y) = a_n x^n y^n + \ldots + a_1 xy + a_o$ for arbitrary values of $a_n, \ldots, a_o$ and n
- $[f](x,y) = a \cdot (x+y)^n$ for arbitrary values of a and n

Lemma 5.4.2

There is no interpretation of f such that

$$C(f) \;\wedge\; [f(f(x,y),z) >_{POL} f(x,f(y,z)) \;\vee\; f(x,f(y,z)) >_{POL} f(f(x,y),z)]$$ ∎

Due to the lemma, it is impossible to guarantee the termination of a system containing the associativity law modulo C w.r.t. a polynomial ordering. In general, the combination of an associativity rule and the commutativity equation is not allowed since it leads to the following infinite derivation (see [Av89]): $f(f(x,y),z) =_C f(z,f(x,y)) > f(f(z,x),y) =_C f(y,f(x,z)) > f(f(y,x),z) =_C f(f(x,y),z)$. Therefore, the underlying theory of the next example consists of both commutativity as well as associativity since an abelian group possesses both properties.

Example 5.4.3 (Abelian group)

$\mathfrak{R}$:

$$x + 0 \rightarrow x$$
$$x + (-x) \rightarrow 0$$

E:

$$x + y = y + x$$
$$(x + y) + z = x + (y + z)$$

I:

$$[0]() = 2$$
$$[-](x) = x^2$$
$$[+](x,y) = xy + x + y$$ ∎

Lemma 5.4.4

$$C_L(f) \quad \text{if} \quad [f](x,y) = \sum_i a_i x^j y^i + (1-j)\cdot x \quad \wedge \quad j \in \{0,1\}$$

$$\text{or} \quad [f](x,y) = (x+1)\cdot\sum_{i\geq 1} a_i y^i + a_0 x + a_0 - 1$$ ■

Special cases:

- $[f](x,y) = a_n xy^n + \ldots + a_1 xy + a_0 x$ for arbitrary values of $a_n,\ldots,a_o$ and n
- $[f](x,y) = a_n y^n + \ldots + a_1 y + x$ for arbitrary values of $a_n,\ldots,a_1$ and n

Example 5.4.5 (see Example 5.4.3)

$\mathfrak{R}$:

$$x + 0 \rightarrow x$$
$$x + (-x) \rightarrow 0$$
$$(x + y) + z \rightarrow x + (y + z)$$

E: $(x + y) + z = (x + z) + y$

I:

$$[0]() = 2$$
$$[-](x) = x^2$$
$$[+](x,y) = xy + x$$

Note that the interpretations for the left commutativity orient the associativity only in one direction. ■

Lemma 5.4.6

$$C_R(f) \quad \text{if} \quad [f](x,y) = \sum_i a_i x^i y^j + (1-j)\cdot y \quad \wedge \quad j \in \{0,1\}$$

$$\text{or} \quad [f](x,y) = (y+1)\cdot\sum_{i\geq 1} a_i x^i + a_0 y + a_0 - 1$$ ■

Special cases:

- $[f](x,y) = a_n x^n y + \ldots + a_1 xy + a_0 y$ for arbitrary values of $a_n,\ldots,a_o$ and n
- $[f](x,y) = a_n x^n + \ldots + a_1 x + y$ for arbitrary values of $a_n,\ldots,a_1$ and n

Example 5.4.7 (see Example 5.4.5)

$$
\begin{aligned}
\Re:\quad & x + 0 && \rightarrow x\\
& x + (-x) && \rightarrow 0\\
& x + (y + z) && \rightarrow (x + y) + z
\end{aligned}
$$

$$
E:\quad x + (y + z) \;=\; y + (x + z)
$$

$$
\begin{aligned}
I:\quad & [0]() && = 2\\
& [-](x) && = x^2\\
& [+](x; y) && = xy + y
\end{aligned}
$$

■

Lemma 5.4.8

- $C(f) \wedge C_L(f)$ if $[f](x, y) = axy + b(x + y) + c$ and $b = c = 0 \vee [a = 0 \wedge b = 1] \vee [a = b \wedge c = b - 1]$
- $C(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$
- $C_L(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$
- $C(f) \wedge C_L(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$ ■

5.5 Distributivity

Lemma 5.5.1

$D_L(f,g)$ if $[f](x,y) = \sum_i a_i x y^i$, $[g](x,y) = b_1 x + b_2 y$

or $[f](x,y) = a_1 xy + a_2(x + y) + a_3$, $[g](x,y) = x + y + b_3$
$\wedge$ $a_3 = b_3(a_1 b_3 - 1)$, $a_2 = a_1 b_3$ ∎

Special cases:
- $[f](x,y) = xy$, $[g](x,y) = b(x + y)$ for arbitrary values of b
- $[f](x,y) = ax(y + 1)$, $[g](x,y) = b(x + y)$ for arbitrary values of a and b
- $[f](x,y) = xy + a(x + y)$, $[g](x,y) = x + y + 1$ for arbitrary values of a

Example 5.5.2 (Boolean algebra)

$\mathfrak{R}$:
$$x + 0 \to x$$
$$x + x \to 0$$
$$x * 0 \to 0$$
$$x * 1 \to x$$
$$x * x \to x$$

E: $(x + y) * z = (x * z) + (y * z)$

I:
$$[0]() = 1$$
$$[1]() = 1$$
$$[+](x,y) = x + y$$
$$[*](x,y) = xy + x$$
∎

Lemma 5.5.3

$D_R(f,g)$ if $[f](x,y) = \sum_i a_i x^i y$, $[g](x,y) = b_1 x + b_2 y$

or $[f](x,y) = a_1 xy + a_2(x + y) + a_3$, $[g](x,y) = x + y + b_3$
$\wedge$ $a_3 = b_3(a_1 b_3 - 1)$, $a_2 = a_1 b_3$ ∎

Special cases:

- $[f](x,y) = xy$, $[g](x,y) = b(x + y)$
  for arbitrary values of b
- $[f](x,y) = ay(x + 1)$, $[g](x,y) = b(x + y)$
  for arbitrary values of a and b
- $[f](x,y) = xy + a(x + y)$, $[g](x,y) = x + y + 1$
  for arbitrary values of a

Analogous with the left and right commutativity, the following axioms cannot be combined:

$$f(f(x,y),z) \rightarrow f(x,f(y,z)) \quad \wedge \quad D_R(f,g)$$

and $f(x,f(y,z)) \rightarrow f(f(x,y),z) \quad \wedge \quad D_L(f,g)$, respectively.

Example 5.5.4

$\mathfrak{R}$: The same rule system as in example 5.5.2

E: $x * (y + z) = (x * y) + (x * z)$

I: The same interpretations as in example 5.5.2 except $[*](x,y) = xy + y$. ■

Lemma 5.5.5

- $D_L(f,g) \quad \wedge \quad D_R(f,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + cy$

  or $[f](x,y) = axy + b(x + y) + c$,
  $[g](x,y) = x + y + d$
  $\wedge \quad c = d(ad - 1)$, $b = ad$

- The following combinations of theories are not allowed, i.e. there exists no polynomial ordering which induces the equality of both sides of the axioms:

  $D_L(f,g) \quad \wedge \quad D_R(g,f)$

  $D_L(f,g) \quad \wedge \quad D_L(g,f)$

  $D_R(f,g) \quad \wedge \quad D_R(g,f)$ ■

5.6 Endomorphism

Lemma 5.6.1

E(f, g) if $[f](x) = x$, $[g](x, y) = \sum_{j,k} a_{jk} x^j y^k$

or $[f](x) = a_1 x + a_2$, $[g](x, y) = b_1 x + b_2 y + b_3$
$\wedge$ $b_3(a_1 - 1) = a_2(b_1 + b_2 - 1)$

or $[f](x) = a_1 x^i$, $[g](x, y) = b_1 x^j y^k$
$\wedge$ $a_1^{j+k-1} = b_1^{i-1}$ ■

Special cases:

- $[f](x) = ax$, $[g](x, y) = bx + cy$
  for arbitrary values of a, b and c
- $[f](x) = x$, $[g](x, y) = ax + by + c$
  for arbitrary values of a, b and c
- $[f](x) = ax + b(a - 1)$, $[g](x, y) = x + y + b$
  for arbitrary values of a and b
- $[f](x) = x^i$, $[g](x, y) = x^j y^k$
  for arbitrary values of i, j and k

In some instances a slightly different endomorphism axiom occurs: f(g(x, y)) = g(f(y), f(x)) where the arguments of g have been exchanged. If the interpretations of the following lemma are given, both sides of the axiom became identical:

Lemma 5.6.2

$[f(g(x, y))] = [g(f(y), f(x))]$ if $[f](x) = x$, $[g](x, y) = \sum_{j,k} a_{jk} x^j y^k$
$\wedge$ $(\forall j, k)\ a_{jk} = a_{kj}$

or $[f](x) = a_1 x + a_2$, $[g](x, y) = b_1(x + y) + b_2$
$\wedge$ $b_2(a_1 - 1) = a_2(2b_1 - 1)$

or $[f](x) = a_1 x^i$, $[g](x, y) = b_1 x^j y^j$
$\wedge$ $a_1^{2j-1} = b_1^{i-1}$ ■

Special cases:

- $[f](x) = ax$, $[g](x, y) = b(x + y)$
  for arbitrary values of a and b
- $[f](x) = x$, $[g](x, y) = a(x + y) + b$
  for arbitrary values of a and b
- $[f](x) = ax + b(a - 1)$, $[g](x, y) = x + y + b$
  for arbitrary values of a and b
- $[f](x) = x^i$, $[g](x, y) = x^j y^j$
  for arbitrary values of i and j

Example 5.6.3 (Group theory)

$$
\begin{array}{lll}
\mathfrak{R}: & 0 + y & \rightarrow & y \\
 & x + 0 & \rightarrow & x \\
 & (-x) + x & \rightarrow & 0 \\
 & x + (-x) & \rightarrow & 0 \\
 & (-x) + (x + y) & \rightarrow & y \\
 & x + ((-x) + y) & \rightarrow & y \\
 & -0 & \rightarrow & 0 \\
 & --x & \rightarrow & x
\end{array}
$$

$$
\begin{array}{llll}
E: & -(x + y) & = & (-y) + (-x) \\
 & (x + y) + z & = & x + (y + z)
\end{array}
$$

$$
\begin{array}{llll}
I: & [0]() & = & 2 \\
 & [-](x) & = & 2x \\
 & [+](x, y) & = & x + y
\end{array}
$$

■

5.7 Idempotency

Note that the idempotency axiom f(x, x) = x can never be part of an underlying theory since it could result in infinite chains (see section 5.1).

Lemma 5.7.1 ([Be86])

There is no interpretation of f such that I(f) is valid. ∎

5.8 Minus

Lemma 5.8.1

$$M(f) \quad \text{if} \quad [f](x) = x$$ ■

Example 5.8.2 (Addition on integers modulo 2)

$$\mathfrak{R}: \quad x + 0 \to x, \quad 0 + y \to y, \quad x + x \to 0$$

$$E: \quad s(s(x)) = x$$

$$I: \quad [0]() = 1, \quad [s](x) = x, \quad [+](x,y) = x + y$$ ■

As stated in [Si89], the real minus theory is a combination of M(f) and a kind of endomorphism axiom (cf. lemma 5.6.2) since they often appear together (see for example groups).

Lemma 5.8.3

$$M(f) \wedge [f(g(x,y))] = [g(f(y),f(x))]$$
$$\text{if} \quad [f](x) = x \;, \quad [g](x,y) = \sum_{i,j} a_{ij} x^i y^j \quad \text{and} \quad (\forall i,j)\; a_{ij} = a_{ji}$$ ■

Note that if we take the usual endomorphism axiom $f(g(x,y)) = g(f(x),f(y))$ instead of the one above, it will make no difference what type of interpretation of the operator g is used.

5.9 Permutativity

Lemma 5.9.1 ([Be86])

P(f) if $[f](x_1,\ldots,x_n) = \Sigma\, a_{i_1\ldots i_n} \cdot x_1^{i_1} \cdot \ldots \cdot x_n^{i_n}$

$\wedge\quad a_{i_1\ldots i_n} = a_{\pi(i_1)\ldots\pi(i_n)}$ ∎

Special cases:

- $[f](x_1,\ldots,x_n) = a_n \cdot x_1^n \ldots x_n^n + \ldots + a_1 \cdot x_1 \ldots x_n + a_o$
  for arbitrary values of $a_n,\ldots,a_o$ and n
- $[f](x_1,\ldots,x_n) = a \cdot (x_1 + \ldots + x_n)^i$
  for arbitrary values of a and i

Example 5.9.2 (Or-exclusive)

$\mathfrak{R}$: $v(x,x,y) \rightarrow x$
$v(y,x,y) \rightarrow y$
$v(x,y,y) \rightarrow y$

E: $v(x,y,z) = v(y,z,x)$

I: $[v](x,y,z) = x + y + z$ ∎

5.10 Transitivity

The operator f of the transitivity axiom

$$f(g(x,y),g(y,z)) = f(g(x,y),g(x,z))$$

is mostly interpreted as the boolean function *and*:

$$g(x,y) \wedge g(y,z) = g(x,y) \wedge g(x,z).$$

Furthermore, a value for the operator g could be any ordering > since it possesses the transitivity property:

$$(x > y) \wedge (y > z) \rightsquigarrow x > z.$$

Lemma 5.10.1 ([Be86])

There are no interpretations of f and g such that $T(f,g)$ is valid. ■

5.11 Unipotency

There are two kinds of unipotency axioms: the left and the right unipotency.

Lemma 5.11.1

$U_L(f,1)$ if $[f](x,y) = x^i y$, $[1]() = 1$ ■

Lemma 5.11.2

$U_R(f,1)$ if $[f](x,y) = xy^i$, $[1]() = 1$ ■

Example 5.11.3 Merging operator

$\mathfrak{R}$: $merge(x \cdot y, u \cdot v) \rightarrow x \cdot merge(y, u \cdot v)$

E: $merge(x, nil) = x$

I: $[nil]() = 1$
$[merge](x,y) = xy$
$[\cdot](x,y) = x + y$ ■

The combination of both unipotency axioms will drastically restrict the class of interpretations of the operator f:

Lemma 5.11.4

$U_L(f,1) \wedge U_R(f,1)$ if $[f](x,y) = xy$, $[1]() = 1$ ■

5.12 Combinations of Theories

We have indicated restrictions in the interpretations of the function symbols that belong to special theories. This section deals with the enumeration of the interpretations of those function symbols which are simultaneously contained in more than one theory. First of all, the combinations of two different theories will be considered.

Lemma 5.12.1

$A(f) \wedge$

- $C(f)$ if $A(f)$
- $C_L(f)$ if $C(f) \wedge C_L(f)$

  Special cases:
  - $[f](x,y) = xy + x + y$
  - $[f](x,y) = xy$
  - $[f](x,y) = x + y$
- $C_R(f)$ if $A(f) \wedge C_L(f)$
- $D_L(f,g)$ if $[f](x,y) = axy \;,\; [g](x,y) = bx + cy$

  or $[f](x,y) = axy + b(x+y) + c \;,\; [g](x,y) = x + y + d$
  $\wedge \; c = d(ad - 1) \;,\; b = ad$
- $D_L(g,f)$ if $[f](x,y) = x + y + a \;,\; [g](x,y) = bxy + c(x+y) + d$
  $\wedge \; d = a(ab - 1) \;,\; c = ab$
- $A(g) \wedge D_L(f,g)$ if $[f](x,y) = axy + b(x+y) + c \;,\; [g](x,y) = x + y + d$
  $\wedge \; c = d(ad - 1) \;,\; b = ad$
- $D_R(f,g)$ if $A(f) \wedge D_L(f,g)$
- $D_R(g,f)$ if $[f](x,y) = x + y \;,\; [g](x,y) = \Sigma a_i x^i y$

  or $A(f) \wedge D_L(g,f)$
- $A(g) \wedge D_R(f,g)$ if $A(f) \wedge A(g) \wedge D_L(f,g)$
- $E(g,f)$ if $[f](x,y) = axy + b(x+y) + c \;,\; [g](x) = x$
  $\wedge \; ac + b = b^2$

or $[f](x,y) = x + y + a$, $[g](x) = bx + c$
$\wedge$ $c = a(b - 1)$

or $[f](x,y) = axy$, $[g](x) = bx^i$
$\wedge$ $b = a^{i-1}$

Special cases:
- $[f](x,y) = x + y$, $[g](x) = ax$
- $[f](x,y) = axy$, $[g](x) = ax^2$

· $U_L(f,1)$ if $U_L(f,1)$ $\wedge$ $U_R(f,1)$

· $U_R(f,1)$ if $U_L(f,1)$ $\wedge$ $U_R(f,1)$ ∎

Lemma 5.12.2

$C(f)$ $\wedge$

· $D_L(f,g)$ if $A(f)$ $\wedge$ $D_L(f,g)$

· $D_L(g,f)$ if $[f](x,y) = a(x+y)$, $[g](x,y) = \Sigma b_i xy^i$
or $A(f)$ $\wedge$ $D_L(g,f)$

· $C(g) \wedge D_L(f,g)$ if $[f](x,y) = axy$, $[g](x,y) = b(x+y)$
or $A(f)$ $\wedge$ $A(g)$ $\wedge$ $D_L(f,g)$

· $D_R(f,g)$ if $A(f)$ $\wedge$ $D_L(f,g)$

· $D_R(g,f)$ if $C(f)$ $\wedge$ $D_L(g,f)$

· $C(g) \wedge D_R(f,g)$ if $C(f)$ $\wedge$ $C(g)$ $\wedge$ $D_L(f,g)$

· $E(g,f)$ if $[f](x,y) = \Sigma a_{jk} x^j y^k$, $[g](x) = x$
$\wedge$ $(\forall j,k)\ a_{jk} = a_{kj}$

or $[f](x,y) = a(x+y) + b$, $[g](x) = cx + d$
$\wedge$ $b(c - 1) = d(2a - 1)$

or $[f](x,y) = ax^i y^i$, $[g](x) = bx^k$
$\wedge$ $b^{2i-1} = a^{k-1}$

Special cases:
- $[f](x,y) = a(x+y)$, $[g](x) = bx$
- $[f](x,y) = axy$, $[g](x) = ax^2$

- $U_L(f,1)$ if $U_L(f,1) \wedge U_R(f,1)$
- $U_R(f,1)$ if $U_L(f,1) \wedge U_R(f,1)$ ∎

Lemma 5.12.3

$C_L(f) \wedge$

- $D_L(f,g)$ if $[f](x,y) = \sum_i a_i x y^i$, $[g](x,y) = bx + cy$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x+y+1$
- $D_L(g,f)$ if $[f](x,y) = x + ay$, $[g](x,y) = \sum b_i x y^i$
- $C_L(g) \wedge D_L(f,g)$ if $[f](x,y) = \sum_i a_i x y^i$, $[g](x,y) = x + by$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x+y+1$
- $D_R(f,g)$ if $A(f) \wedge D_L(f,g)$
- $D_R(g,f)$ if $[f](x,y) = x + ay$, $[g](x,y) = \sum b_i x^i y$
  or $A(f) \wedge D_L(g,f)$
- $C_L(g) \wedge D_R(f,g)$ if $[f](x,y) = axy$, $[g](x,y) = x + by$
  or $A(f) \wedge A(g) \wedge D_L(f,g)$
- $E(g,f)$ if $[f](x,y) = \sum a_i x^j y^i + (1 - j)x$, $[g](x) = x$ $\wedge\ j \in \{0,1\}$
  or $[f](x,y) = (x+1)\sum_{i\geq 1} a_i y^i + a_0 x + a_0 - 1$, $[g](x) = x$
  or $[f](x,y) = x + ay + b$, $[g](x) = cx + d$ $\wedge\ b(c - 1) = ad$
  or $[f](x,y) = axy^i$, $[g](x) = bx^k$ $\wedge\ b^i = a^{k-1}$

Special cases: - $[f](x,y) = x + ay$, $[g](x) = bx$
- $[f](x,y) = axy$, $[g](x) = ax^2$

- $U_L(f,1)$ if $U_L(f,1) \wedge U_R(f,1)$

- $U_R(f,1)$ if $U_R(f,1)$ ∎

Lemma 5.12.4

$C_R(f) \wedge$

- $D_L(f,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + cy$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x+y+1$

- $D_L(g,f)$ if $[f](x,y) = ax + y$, $[g](x,y) = \Sigma b_i xy^i$
  or $A(f) \wedge D_L(g,f)$

- $C_R(g) \wedge D_L(f,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + y$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x+y+1$

- $D_R(f,g)$ if $[f](x,y) = \Sigma a_i x^i y$, $[g](x,y) = bx + cy$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x+y+1$

- $D_R(g,f)$ if $[f](x,y) = ax + y$, $[g](x,y) = \Sigma b_i x^i y$
  or $A(f) \wedge D_L(g,f)$

- $C_R(g) \wedge D_R(f,g)$ if $[f](x,y) = \Sigma a_i x^i y$, $[g](x,y) = bx + y$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x+y+1$

- $E(g,f)$ if $[f](x,y) = \Sigma a_i x^i y^j + (1 - j)y$
  $\wedge \; j \in \{0,1\}$
  or $[f](x,y) = (y+1)\sum_{i\geq 1} a_i x^i + a_0 y + a_0 - 1$, $[g](x) = x$
  or $[f](x,y) = ax + y + b$, $[g](x) = cx + d$
  $\wedge \; b(c - 1) = ad$

or $[f](x,y) = ax^i y \;,\; [g](x) = bx^k$
$\wedge \quad b^i = a^{k-1}$

Special cases:
- $[f](x,y) = ax + y \;,\; [g](x) = bx$
- $[f](x,y) = axy \;,\; [g](x) = ax^2$

- $U_L(f,1)$ if $U_L(f,1)$
- $U_R(f,1)$ if $U_L(f,1) \wedge U_R(f,1)$ ∎

In the remaining part of this section we use $\lightning$ if the given axioms cannot represent an underlying theory (since the polynomial ordering fails).

Lemma 5.12.5

$D_L(f,g) \wedge$

- $E(h,f)$ if $[f](x,y) = \Sigma a_i xy^i \;,\; [g](x,y) = bx + cy \;,\; [h](x) = x$

  or $[f](x,y) = axy^i \;,\; [g](x,y) = bx + cy \;,\; [h](x) = dx^k$
  $\wedge \quad d^i = a^{k-1}$

  or $[f](x,y) = axy + b(x + y) + c \;,\; [g](x,y) = x + y + d \;,$
  $[h](x) = x \quad \wedge \quad c = d(ad - 1) \;,\; b = ad$

- $E(h,g)$ if $[f](x,y) = \Sigma\, a_i xy^i \;,\; [g](x,y) = bx + cy \;,\; [h](x) = dx$

  or $[f](x,y) = axy + b(x + y) + c \;,\; [g](x,y) = x + y + d \;,$
  $[h](x) = px + q \quad \wedge \quad c = d(ad - 1) \;,\; b = ad \;,\; d(p - 1) = q$

- $U_L(f,1)$ if $U_L(f,1) \wedge U_R(f,1) \wedge [g](x,y) = ax + by$
- $U_L(g,1)$ $\lightning$
- $U_R(f,1)$ if $U_R(f,1) \wedge [g](x,y) = ax + by$
- $U_R(g,1)$ $\lightning$ ∎

Lemma 5.12.6

$D_R(f,g) \wedge$

- $E(h,f)$ if $[f](x,y) = \Sigma a_i x^i y \;,\; [g](x,y) = bx + cy \;,\; [h](x) = x$

or $[f](x,y) = ax^iy$, $[g](x,y) = bx + cy$, $[h](x) = dx^k$
$\wedge$ $d^i = a^{k-1}$

or $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$,
$[h](x) = x$ $\wedge$ $c = d(ad - 1)$, $b = ad$

- $E(h,g)$ if $[f](x,y) = \Sigma a_i x^i y$, $[g](x,y) = bx + cy$, $[h](x) = dx$

  or $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$,
  $[h](x) = px + q$ $\wedge$ $c = d(ad - 1)$, $b = ad$, $d(p - 1) = q$

- $U_L(f,1)$ if $U_L(f,1)$ $\wedge$ $[g](x,y) = ax + by$

- $U_L(g,1)$ $\not\!\!\!\;$ ↯

- $U_R(f,1)$ if $U_L(f,1)$ $\wedge$ $U_R(f,1)$ $\wedge$ $[g](x,y) = ax + by$

- $U_R(g,1)$ ↯ ∎

Lemma 5.12.7

$E(f,g)$ $\wedge$

- $M(f)$ if $[f](x) = x$, $[g](x,y) = \Sigma\, a_{jk} x^j y^k$

- $U_L(g,1)$ if $U_L(g,1)$ $\wedge$ $[f](x) = x^k$

- $U_R(g,1)$ if $U_R(g,1)$ $\wedge$ $[f](x) = x^k$ ∎

For compactness, the combinations of three different theories are presented in a compressed way, i.e. an algorithm (as inference rules) is given to compute the interpretations. This algorithm is based on all lemmata of this chapter. The results of these lemmata will be used as rules (axioms) $\mathfrak{A}$, e.g.

- $A(f)$ $\wedge$ $C(f)$ if $A(f)$ will be transformed into $A(f)$ $\wedge$ $C(f) \rightarrow A(f)$
- $C_L(f)$ $\wedge$ $D_L(g,f)$ if $[f](x,y) = x + ay$, $[g](x,y) = \Sigma b_i xy^i$ will be transformed into $C_L(f) \wedge D_L(g,f) \rightarrow [f](x,y) = x + ay \wedge [g](x,y) = \Sigma b_i xy^i$.

For computing the restricted interpretations of all combinations of three theories, the results of the following lemma have to be added to the axioms $\mathfrak{A}$.

Lemma 5.12.8

- $A(f) \wedge C_L(f) \wedge E(g,f)$ if $[f](x,y) = axy + b(x + y) + c$, $[g](x) = x$ and $[b = c = 0] \vee [a = b , c = b - 1]$

  or $[f](x,y) = x + y + a$, $[g](x) = bx + c \wedge c = a(b - 1)$

  or $[f](x,y) = axy$, $[g](x) = bx^i \wedge b = a^{i-1}$

  Special cases:
  - $[f](x,y) = x + y$, $[g](x) = ax$
  - $[f](x,y) = axy$, $[g](x) = ax^2$

- $A(f) \wedge D_L(f,g) \wedge E(h,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + cy$, $[h](x) = dx$

  or $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$, $[h](x) = px + q \wedge c = d(ad - 1)$, $b = ad$, $d(p - 1) = q$

- $A(f) \wedge D_L(f,g) \wedge E(h,f)$ if $[f](x,y) = axy$, $[g](x,y) = bx + cy$ $[h](x) = dx^i \wedge d = a^{i-1}$

  or $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$, $[h](x) = x \wedge c = d(ad - 1)$, $b = ad$

- $A(f) \wedge A(g) \wedge D_L(f,g) \wedge E(h,g)$ if $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$, $[h](x) = px + q \wedge c = d(ad - 1)$, $b = ad$, $d(p - 1) = q$

- $A(f) \wedge D_L(f,g) \wedge U_L(f,1)$ if $[f](x,y) = xy$, $[g](x,y) = ax + by$, $[1]() = 1$

- $A(f) \wedge D_L(f,g) \wedge U_R(f,1)$ if $A(f) \wedge D_L(f,g) \wedge U_L(f,1)$

- $A(f) \wedge D_L(g,f) \wedge U_L(g,1)$ if $[f](x,y) = x+y$, $[g](x,y) = xy$, $[1]() = 1$

- $A(f) \wedge D_L(g,f) \wedge U_R(g,1)$ if $[f](x,y) = x+y$, $[g](x,y) = xy^i$, $[1]() = 1$

- $A(f) \wedge D_R(g,f) \wedge E(h,f)$ if $[f](x,y) = x+y$, $[g](x,y) = \Sigma a_i x^i y$, $[h](x) = bx$

  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $A(f) \wedge E(g,f) \wedge M(g)$ if $[f](x,y) = axy + b(x + y) + c$, $[g](x) = x \wedge ac + b = b^2$

- $C(f) \wedge D_L(g,f) \wedge E(h,f)$ if $[f](x,y) = a(x + y)$, $[g](x,y) = \Sigma b_i xy^i$, $[h](x) = cx$

  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $C(f) \wedge C(g) \wedge D_L(f,g) \wedge E(h,g)$ if $[f](x,y) = axy$, $[g](x,y) = b(x+y)$, $[h](x) = cx$

  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $C(f) \wedge D_L(g,f) \wedge U_L(g,1)$ if $[f](x,y) = a(x + y)$, $[g](x,y) = xy$, $[1]() = 1$

- $C(f) \wedge D_L(g,f) \wedge U_R(g,1)$ if $[f](x,y) = a(x + y)$, $[g](x,y) = xy^i$, $[1]() = 1$

- $C(f) \wedge D_R(g,f) \wedge E(h,f)$ if $[f](x,y) = a(x + y)$, $[g](x,y) = \Sigma b_i x^i y$, $[h](x) = cx$

  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $C(f) \wedge C(g) \wedge D_R(f,g) \wedge E(h,g)$ if $[f](x,y) = axy$, $[g](x,y) = b(x+y)$, $[h](x) = cx$

or $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$, $[h](x) = px + q \wedge c = d(ad - 1)$, $b = ad$, $d(p - 1) = q$

- $C[f] \wedge E[g,f] \wedge M[g]$ if $[f](x,y) = \Sigma a_{jk} x^j y^k$, $[g](x) = x \wedge (\forall j,k)\ a_{jk} = a_{kj}$

- $C_L[f] \wedge C_L[g] \wedge C_R[f] \wedge C_R[g] \wedge D_L[f,g]$ if $[f](x,y) = axy$, $[g](x,y) = x + y$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x + y + 1$

- $C_L[f] \wedge C_L[g] \wedge C_R[f] \wedge C_R[g] \wedge D_R[f,g]$ if $C_L[f] \wedge C_L[g] \wedge C_R[f] \wedge C_R[g] \wedge D_L[f,g]$

- $C_L[f] \wedge D_L[g,f] \wedge E[h,f]$ if $[f](x,y) = x + ay$, $[g](x,y) = \Sigma b_i x y^i$, $[h](x) = cx$
  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $C_L[f] \wedge C_L[g] \wedge D_L[f,g] \wedge E[h,g]$ if $[f](x,y) = \Sigma a_i x y^i$, $[g](x,y) = x + by$, $[h](x) = cx$
  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x + y + 1$, $[h](x) = px + p - 1$

- $C_L[f] \wedge D_L[g,f] \wedge U_L[g,1]$ if $[f](x,y) = x + ay$, $[g](x,y) = xy$, $[1]() = 1$

- $C_L[f] \wedge D_L[g,f] \wedge U_R[g,1]$ if $[f](x,y) = x + ay$, $[g](x,y) = xy^i$, $[1]() = 1$

- $C_L[f] \wedge D_R[g,f] \wedge E[h,f]$ if $[f](x,y) = x + ay$, $[g](x,y) = \Sigma b_i x^i y$, $[h](x) = cx$
  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $C_L(f) \wedge C_L(g) \wedge D_R(f,g) \wedge E(h,g)$ if $[f](x,y) = axy$, $[g](x,y) = x + by$, $[h](x) = cx$

  or $[f](x,y) = axy + b(x + y) + c$, $[g](x,y) = x + y + d$, $[h](x) = px + q \wedge c = d(ad - 1)$, $b = ad$, $d(p - 1) = q$

- $C_L(f) \wedge E(g,f) \wedge M(g)$ if $[f](x,y) = \Sigma\, b_i x^j y^i + (1-j)x$, $[g](x) = x \wedge j \in \{0,1\}$

  or $[f](x,y) = (x+1) \sum_{i \geq 1} a_i y^i + a_0 x + a_0 - 1$, $[g](x) = x$

- $C_R(f) \wedge D_L(g,f) \wedge E(h,f)$ if $[f](x,y) = ax + y$, $[g](x,y) = \Sigma b_i x y^i$, $[h](x) = cx$

  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $a(p - 1) = q$

- $C_R(f) \wedge C_R(g) \wedge D_L(f,g) \wedge E(h,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + y$, $[h](x) = cx$

  or $[f](x,y) = axy + a(x + y) + a - 1$, $[g](x,y) = x + y + 1$, $[h](x) = bx + b - 1$

- $C_R(f) \wedge D_L(g,f) \wedge U_L(g,1)$ if $[f](x,y) = ax + y$, $[g](x,y) = xy$, $[1]() = 1$

- $C_R(f) \wedge D_L(g,f) \wedge U_R(g,1)$ if $[f](x,y) = ax + y$, $[g](x,y) = xy^i$, $[1]() = 1$

- $C_R(f) \wedge D_R(g,f) \wedge E(h,f)$ if $[f](x,y) = ax + y$, $[g](x,y) = \Sigma b_i x^i y$, $[h](x) = cx$

  or $[f](x,y) = x + y + a$, $[g](x,y) = bxy + c(x + y) + d$, $[h](x) = px + q \wedge d = a(ab - 1)$, $c = ab$, $q = a(p - 1)$

- $C_R(f) \wedge C_R(g) \wedge D_L(f,g) \wedge E(h,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + y$, $[h](x) = cx$

or $[f](x,y) = axy + a(x + y) + a - 1$,
$[g](x,y) = x + y + 1$,
$[h](x) = px + p - 1$

- $C_R[f] \wedge C_R[g] \wedge D_R[f,g] \wedge E[h,g]$ if $[f](x,y) = \Sigma a_i x^i y$,
$[g](x,y) = bx + y$, $[h](x) = cx$

or $[f](x,y) = axy + a(x + y) + a - 1$,
$[g](x,y) = x + y + 1$,
$[h](x) = bx + b - 1$

- $C_R[f] \wedge E[g,f] \wedge M[g]$ if $[f](x,y) = \Sigma a_i x^i y^j + (1-j)y$,
$[g](x) = x \wedge j \in \{0,1\}$

or $[f](x,y) = (y+1) \sum_{i \geq 1} a_i x^i + a_0 y + a_0 - 1$, $[g](x) = x$

- $D_L[f,g] \wedge E[h,g] \wedge M[h]$ if $[f](x,y) = \Sigma a_i x y^i$,
$[g](x,y) = bx + cy$, $[h](x) = x$

or $[f](x,y) = axy + b(x + y) + c$,
$[g](x,y) = x + y + d$, $[h](x) = x$
$\wedge$ $c = d(ad - 1)$, $b = ad$

- $D_R[f,g] \wedge E[h,g] \wedge M[h]$ if $[f](x,y) = \Sigma a_i x^i y$,
$[g](x,y) = bx + cy$, $[h](x) = x$

or $[f](x,y) = axy + b(x + y) + c$,
$[g](x,y) = x + y + d$, $[h](x) = x$
$\wedge$ $c = d(ad - 1)$, $b = ad$ ∎

Algorithm 5.12.9

This algorithm computes the restrictions for the interpretations of operators concerning special theories (combinations of up to three different theories). The input data are

- the specification of the theories, e.g. $A[f] \wedge C[f]$ and
- the axioms of all lemmata contained in this chapter, denoted by $\mathfrak{A}$.

The inference rules (cf. algorithm 3.3.13) are given as follows:

(1) Removing duplicates

$$\frac{\bigwedge_{i \in I} X_i \;\wedge\; X_j \;,\; \mathfrak{A}}{\bigwedge_{i \in I} X_i \;,\; \mathfrak{A}} \qquad \text{if} \quad j \in I$$

(2) Reducing with rules of $\mathfrak{A}$

$$\frac{\bigwedge_{i \in I} X_i \;\wedge\; \bigwedge_{i \in J} X_i \;,\; \mathfrak{A}}{\sigma(r) \;\wedge\; \bigwedge_{i \in J} X_i \;,\; \mathfrak{A}} \qquad \text{if} \quad \bigwedge_{i \in I} Y_i \rightarrow r \in \mathfrak{A} \;\wedge\; (\exists \sigma, \pi)(\forall i \in I) \;\; \sigma(X_i) = Y_{\pi(i)}$$

where σ is a substitution on function symbols, e.g. $\sigma = \{f \leftarrow g\}$ and $\sigma(U(f,1)) = U(g,1)$, $\sigma([f](x,y) = xy \wedge [1]() = 1) = ([g](x,y) = xy \wedge [1]() = 1)$. ∎

Example 5.12.10

An application of the algorithm to the theory

$$A(f) \;\wedge\; A(g) \;\wedge\; C_L(g) \;\wedge\; D_R(g,f)$$

could produce the following steps:

$$\frac{(C_L(g) \wedge D_R(g,f)) \;\wedge\; (A(f) \wedge A(g)) \;,\; \mathfrak{A}}{(A(g) \wedge D_L(g,f)) \;\wedge\; (A(f) \wedge A(g)) \;,\; \mathfrak{A}}$$

since $\mathfrak{A}$ contains the rule $C_L(f) \wedge D_R(f,g) \rightarrow A(f) \wedge D_L(f,g)$ (lemma 5.12.3) and there is a σ (= $\{f \leftarrow g \;,\; g \leftarrow f\}$, renaming the operators) and a π (= id) such that $\sigma(C_L(g) \wedge D_R(g,f)) = (C_L(f) \wedge D_R(f,g))$.

$$\frac{A(g) \;\wedge\; D_L(g,f) \;\wedge\; A(f) \;\wedge\; A(g) \;,\; \mathfrak{A}}{A(f) \;\wedge\; A(g) \;\wedge\; D_L(g,f) \;,\; \mathfrak{A}}$$

because of applying the inference rule (1) which removes one of the two A(g)'s.

$$\frac{A(f) \;\wedge\; A(g) \;\wedge\; D_L(g,f) \;,\; \mathfrak{A}}{[f](x,y) = x+y+d \;\wedge\; [g](x,y) = axy + b(x+y)+c \;\wedge\; c = d(ad-1) \;\wedge\; b = ad, \; \mathfrak{A}}$$

since the rule $A(f) \wedge A(g) \wedge D_L(f,g) \rightarrow ([f](x,y) = x+y+d \wedge [g](x,y) = axy + b(x+y)+c \wedge c = d(ad-1) \wedge b = ad)$ of lemma 5.12.1 can be applied to $A(f) \wedge A(g) \wedge D_L(g,f)$ after substituting f and g. ∎

Lemma 5.12.11

The algorithm 5.12.9 always terminates. It provides restrictions for interpretations that have to be met by combinations of up to three different theories. ∎

Obviously, in order to use the algorithm for combinations of more than three theories we have to extend $\mathfrak{A}$ by lemmata similar to lemma 5.12.8.

5.13 Special Theories

Finally, some theories which frequently occur in practice will be examined. We deal with variable-reducing equations, subterm-minimizing equations, some kinds of if-then-else equality and special algebraic structures including abelian groups, quasi-groups and rings.

Definition 5.13.1 Variable-reducing axioms

An equation $s = t$ is called variable-reducing
iff $V(s) \subset V(t) \vee V(t) \subset V(s)$ ∎

Thus, an equation is called variable-reducing if the *set* of variables of one side is *properly* included in that of the other side. For example, $x/x = 1$ is variable-reducing.

Lemma 5.13.2

There are no interpretations of function symbols such that both sides of a variable-reducing equation are equivalent w.r.t. a polynomial ordering. ∎

A more general result in regard to arbitrary E-termination is the fact that the set of variables of both sides of the axioms must be identical (see 5.1).

We now consider axioms where one side is a special subterm of the other one.

Definition 5.13.3 Subterm-minimizing axioms

An equation $s = t$ is called subterm-minimizing
iff
- t is a proper subterm of s and
- s contains at least one function symbol with more than one argument outside t ∎

For example, $(-x) + 0 = -x$ is subterm-minimizing whereas $-(-(x + x)) = x + x$ is not.

Lemma 5.13.4

There are no interpretations of function symbols such that both sides of a subterm-minimizing equation are equivalent w.r.t. a polynomial ordering. ∎

The more general assertion about subterms (s = t where t is a subterm of s) is not true since the use of the identity as an interpretation of unary function symbols would imply the equivalence (e.g., $[-(-(x + x))] = [x + x]$ if $[-](x) = x$).

A range of applications for the two general conditions above (5.13.2 and 5.13.4) can be found in regard to various axiomizations of the if-then-else construct. The usual specification of this operator consists of the two simple rules if(true, x, y) → x, if(false, x, y) → y.

Lemma 5.13.5

There are no interpretations of the function symbol *if* such that both sides of each of the following equations are equivalent w.r.t. polynomial orderings:

$$\begin{aligned}
&\text{if}(x, y, y) &&= y\\
&\text{if}(x, x, \text{false}) &&= x\\
&\text{if}(x, \text{true}, x) &&= x\\
&\text{if}(x, \text{if}(x, y, z), z) &&= \text{if}(x, y, z)\\
&\text{if}(x, y, \text{if}(x, y, z)) &&= \text{if}(x, y, z)
\end{aligned}$$ ∎

The axioms of the last lemma demonstrate the difficulty of using theorems about if-then-else as underlying theories. Common to all these axioms is the fact that one side contains one more if-operator than the other side. An if-then-else axiom will be more suitable for an underlying theory if the number of occurrences of the if-operator is equal for both sides (see the next two lemmata).

Lemma 5.13.6

$$[\text{if}(\text{if}(x, y, z), u, v)] = [\text{if}(x, \text{if}(y, u, v), \text{if}(z, u, v))]$$

if

$$\begin{aligned}
&[\text{if}](x, y, z) = a_1xy + a_2xz + a_3x + a_4y + a_5z + a_6\\
&\wedge \quad \cdot\ a_1, a_2 > 0 \quad \wedge \quad a_3 = a_4 = a_5 = a_6 = 0\\
&\qquad \text{or} \quad \cdot\ a_3, a_4, a_5, a_6 > 0 \quad \wedge\\
&\qquad\qquad a_3 = a_4 + a_5\\
&\qquad\qquad a_1a_5 = a_2a_4\\
&\qquad\qquad a_1a_6 = a_4(a_4 + a_5 - 1)
\end{aligned}$$ ∎

Note that there is no separate interpretation which induces the equivalence of the two terms of lemma 5.13.6. Special cases of the admissible interpretations are for example, $ax(y + z)$ and $(x + 1)(y + z + 2) - 1$.

Lemma 5.13.7

[if(x, if(y, z, u), if(y, v, w))] = [if(y, if(x, z, v), if(x, u, w))]

if

$[if](x, y, z) = \sum_i (a_i x^i y + b_i x^i z) + c_1 y$ ∎

For example, x(ay + bz) is a special case of the interpretations of lemma 5.13.7. Analogous with lemma 5.13.6, there exists no separate interpretation causing the equivalence of both terms (cf. suggestion 3.2.10). The union of both if-then-else axioms (5.13.6 and 5.13.7) can be used as an underlying theory if the interpretation of the if-operator has the form x(ay + bz).

We conclude this chapter by examining the potential of some algebraic structures of being underlying theories w.r.t. polynomial orderings. The test includes boolean rings, quasi-groups and abelian groups.

Definition 5.13.8 Abelian group

The following system specifies an abelian group:

$\mathfrak{R}_{AG}$:			
	x + 0	=	x
	(x + y) + z	=	x + (y + z)
	x + y	=	y + x
	i(0)	=	0
	i(i(x))	=	x
	i(x + y)	=	i(y) + i(x)
	x + i(x)	=	0 ∎

Due to lemma 5.13.2, the variable-reducing axiom of the existence of inverse elements (x + i(x) = 0) cannot be used as a part of an underlying theory. If this axiom is eliminated from $\mathfrak{R}_{AG}$, the following interpretations will suffice:

Lemma 5.13.9

Both sides of the axioms belonging to $\mathfrak{R}_{AG}\backslash\{x + i(x) = 0\}$ are equivalent w.r.t. the polynomial ordering based on the interpretations

[+](x, y)	=	xy
[i](x)	=	x
[0]( )	=	1 ∎

Now, we study the theory of quasi-groups based on the three binary operators $*$, $\backslash$ and $/$. The system of the following lemma represents a canonical specification of this theory (see [Hu80a]).

Lemma 5.13.10 Quasi-group

There are no interpretations of function symbols such that both sides of the axioms of a quasi-group

$$\begin{aligned} x * (x \backslash y) &\rightarrow y \\ (x / y) * y &\rightarrow x \\ x \backslash (x * y) &\rightarrow y \\ (x * y) / y &\rightarrow x \\ (x / y) \backslash x &\rightarrow y \\ x / (y \backslash x) &\rightarrow y \end{aligned}$$

are equivalent w.r.t. a polynomial ordering. ∎

Note that a ring is an abelian group together with a multiplicative operation $*$ such that the elements possess an additional semigroup structure (associativity law of $*$). Moreover, the multiplication is distributive w.r.t. the addition. Thus, the set of operators consists of three elements: $+$, unary $-$ and $*$.

Definition 5.13.11 Ring

$$\begin{aligned} \mathfrak{R}_R: \quad & \mathfrak{R}_{AG} \\ \cup \quad & \{x * (y * z) = (x * y) * z \\ & \;\; x * (y + z) = (x * y) + (x * z) \\ & \;\; (x + y) * z = (x * z) + (y * z)\} \end{aligned}$$

specifies a ring. ∎

It is impossible to consider $\mathfrak{R}_R$ as an underlying theory if polynomial orderings are used since the distributivity axioms require a separate interpretation of $+$ (cf. lemma 5.5.5) which is a contradiction to the choice $[+](x,y) = xy$ due to $\mathfrak{R}_{AG}$ (see lemma 5.13.9). Obviously, a more special ring - the boolean ring $\mathfrak{R}_{BR}$ - cannot be an underlying theory, either. Finally, we will investigate the axioms that have to be eliminated in $\mathfrak{R}_{BR}$ if it is used as an underlying theory.

Definition 5.13.12 Boolean ring

The theory of a boolean ring contains the following axioms:

$$
\begin{array}{llrcl}
\mathfrak{R}_{BR}: & \mathfrak{R}_R & & & \\
 & \cup & \{x * y & = & y * x \\
 & & x + x & = & x \\
 & & x * x & = & x \\
 & & x + (x * y) & = & x \\
 & & x * (x + y) & = & x \\
 & & x + (y * z) & = & (x + y) * (x + z) \\
 & & (x * y) + z & = & (x + z) * (y + z) \\
 & & i(x + y) & = & i(x) * i(y) \\
 & & i(x * y) & = & i(x) + i(y) \\
 & & x * i(x) & = & 1\}
\end{array}
$$

■

The axioms that are either variable-reducing or subterm-minimizing can immediately be excluded (see lemmata 5.13.2 and 5.13.4). These axioms are: the idempotency ($x + x = x$ and $x * x = x$), the adsorption ($x + (x * y) = x$ and $x * (x + y) = x$) and the complement ($x + i(x) = 0$ and $x * i(x) = 1$). The axioms about the distributivity $D_L(+,*) \wedge D_R(+,*) \wedge D_L(*,+) \wedge D_R(*,+)$ are the most difficult to handle. Due to lemma 5.5.5, either $D_L(+,*) \wedge D_R(+,*)$ must be part of the underlying theory and the remainder must be oriented, or vice versa. However, both possibilities are not realizable. Thus, all distributivity axioms have to be part of the reducing system. Unfortunately, these equations cannot be oriented in the usual direction with any polynomial ordering (the proof of this statement is based on suggestion 3.2.10 which induces the interpretations of + and * to be mixed polynomials). This leads to the following lemma.

Lemma 5.13.13

It will be impossible to use any part of the boolean ring $\mathfrak{R}_{BR}$ as an underlying theory (and the remaining part as reduction rules), if the termination is to be proved by a polynomial ordering. ■

Based on the assertion above, it is also hopeless to orient all axioms of $\mathfrak{R}_{BR}$ in the usual way w.r.t. a polynomial ordering.

5.14 Discussion

Term rewriting systems using underlying theories are very beneficial. In order to use such systems effectively they must be terminating w.r.t. the underlying theory. This special termination property can be guaranteed by polynomial orderings based on restricted polynomials which imply the equality (w.r.t. the polynomial ordering) of equal (w.r.t. the theory) terms.

The present chapter discussed various theories including associativity, associativity and commutativity, associativity and commutativity and distributivity, idempotency, permutativity as well as transitivity. The appropriate polynomials for these theories are presented in [Be86].

Moreover, some other theories such as left commutativity $(f(f(x,y),z) = f(f(x,z),y))$, right commutativity $(f(x,f(y,z)) = f(y,f(x,z)))$, left and right distributivity, endomorphism, minus theory $(f(f(x)) = x \wedge f(g(x,y)) = g(f(y),f(x)))$ and left and right unipotency were presented.

Section 5.12 listed special classes of polynomials that are useful for the combinations of the theories described in former sections.

Finally, some more specific theories were examined. Among others, we proved that there are no interpretations of operators such that the sides of a variable-reducing equation as well as a subterm-minimizing (a special subterm property) equation are equivalent w.r.t. a polynomial ordering. We applied these assertions to if-then-else axioms and algebraic structures and pointed out that a lot of those equations are unsuitable as underlying theories.

Conclusively, we would like to remark that many of the restricted classes of polynomials presented are complete ones, i.e. the if-assertion of the lemmata can be replaced by if-and-only-if. For compactness, we have skipped the corresponding proofs.

6 An Improved Polynomial Ordering

Polynomial orderings provide a powerful technique for proving the termination as well as the E-termination of rewrite systems. Naturally, there exist equations which cannot be oriented with their help. In this chapter we present and bring forward arguments that support an extension of the polynomial orderings developed by Lankford ([La79]). It is based on the approach of the Knuth-Bendix ordering by using an additional ordering on the function symbols. We will improve this ordering by incorporating a status function.

6.1 Motivation

First of all we point out the limits of the polynomial orderings. Let us have a look at two basic examples, for which it is not difficult to deduce appropriate interpretations.

i) The axiomatic definition of the addition on natural numbers

$$x + 0 \rightarrow x$$
$$x + s(y) \rightarrow s(x + y)$$

is terminating since the following interpretations orient the rules in the desired way:

$$[+](x, y) = x + 2y \ , \ [s](x) = x + 1 \ , \ [0](\ ) = 2$$

ii) The termination of the following definition of the multiplication on natural numbers

$$x * 0 \rightarrow 0$$
$$x * s(y) \rightarrow x + (x * y)$$

can be guaranteed by using the polynomial interpretations

$$[*](x, y) = xy \ , \ [+](x, y) = x + y \ , \ [s](x) = 2x + 1 \ , \ [0](\ ) = 2$$

Note that the second example is not a complete definition of * since + is not defined. Therefore, we have to combine i) and ii):

Example 6.1.1

The axiomatic definition of the multiplication

$$x * 0 \rightarrow 0$$
$$x * s(y) \rightarrow x + (x * y)$$
$$x + 0 \rightarrow x$$
$$x + s(y) \rightarrow s(x + y)$$

cannot be oriented with the help of any polynomial ordering. For the simple reason that [s] must be of the form $x + c$ to orient the last rule and it has to have the form $ax + c$ with $a > 1$ for guaranteeing the termination of the second rule. ∎

In the history of polynomial orderings there have already been two extensions. The first one concerns the classes of functions used as interpretations. For example, it is possible to include exponential functions while preserving the main conditions (see chapter 2). With this approach, it is possible to orient the rules of example 6.1.1:

Let
$$[*](x,y) = 2^{xy}$$
$$[+](x,y) = x + 2y$$
$$[s](x) = x + 1$$
$$[0]() = 2$$

Then,

$x * 0 > 0$	since	$2^{2x} >_P 2$
$x * s(y) > x + (x * y)$	since	$2^{xy} \cdot 2^{x} >_P 2^{xy} \cdot 2 + x$
$x + 0 > x$	since	$x + 4 >_P x$
$x + s(y) > s(x + y)$	since	$x + 2y + 2 >_P x + 2y + 1$

The second improvement of the original polynomial ordering is based on the concept of the concatenation of well-founded orderings. In [De83] a convenient set of basic rules is provided by which well-founded orderings can be constructed. One of these principles implies that a lexicographic ordering of fixed-length tuples is well-founded if the orderings on the components are. In other words, the lexicographic concatenation of well-founded orderings $>_1, \ldots, >_n$,

$$
\begin{aligned}
s > t \quad \text{iff} \quad & s >_1 t \\
\text{or} \quad & s =_1 t \;\wedge\; s >_2 t \\
\text{or} \quad & s =_1 t \;\wedge\; s =_2 t \;\wedge\; s >_3 t \\
& \vdots \\
\text{or} \quad & s =_1 t \;\wedge \ldots \wedge\; s =_{n-1} t \;\wedge\; s >_n t
\end{aligned}
$$

is well-founded itself. In the remaining part of this chapter we use the following compressed representation of the definition above:

$$
\begin{aligned}
s > t \quad \text{iff} \quad & - \; s >_1 t \\
& - \; s >_2 t \\
& \vdots \\
& - \; s >_n t
\end{aligned}
$$

The concept of the concatenation of orderings has been applied to the polynomial ordering in such a way that several different polynomial orderings are concatenated, i.e. $>_1, \ldots, >_n$ are polynomial orderings with distinct interpretations. This technique is called polynomial ordering with interpretations by a Cartesian product of polynomials (see for example [BL87]). Note that with this approach we are also able to guarantee the termination of the rule system of example 6.1.1:

> Let $>_1 = >_{POL}$ using the interpretations $[*](x,y) = xy$, $[+](x,y) = x + y$, $[s](x) = x + 2$ and $[0](\,) = 2$.
> Let $>_2 = >_{POL}$ with $[*](x,y) = xy$, $[+](x,y) = xy$, $[s](x) = x + 2$ and $[0](\,) = 2$ as polynomial interpretations.
> Then,
>
> $$\begin{array}{lll} x * 0 & >_1 & 0 \\ x * s(y) & >_1 & x + (x * y) \\ x + 0 & >_1 & x \\ x + s(y) & =_1 & s(x + y) \quad \wedge \quad x + s(y) >_2 s(x + y) \end{array}$$

The next section deals with another improvement of the polynomial ordering which is also based on the technique of concatenating well-founded orderings.

6.2 Polynomial Ordering with Status

In order to define another extension of the polynomial ordering we need an appropriate formalism outlined below.

Note that a term ordering $\succ$ is used to compare terms. Since operators have terms as arguments we define an extension of $\succ$, called lexicographically greater ($\succ^{lex}$), on tuples of terms as follows:

$$(s_1, s_2, \ldots, s_m) \succ^{lex} (t_1, t_2, \ldots, t_n)$$
$$\text{if either } m > 0 \wedge n = 0$$
$$\text{or } s_1 \succ t_1$$
$$\text{or } s_1 = t_1 \wedge (s_2, \ldots, s_m) \succ^{lex} (t_2, \ldots, t_n)$$

If there is no order of succession among the terms of such tuples then the structures are called multisets. Multisets differ from sets by allowing multiple occurrences of identical elements. The multiset difference is represented by \. The extension of $\succ$ on multisets of terms is defined as follows. A multiset S is greater than a multiset T, denoted by

$$S \succ\succ T$$
$$\text{iff} \cdot S \neq T \wedge$$
$$\cdot (\forall t \in T\backslash S)(\exists s \in S\backslash T) \; s \succ t$$

i.e. $S \succ\succ T$ if T can be obtained from S by replacing one or more terms in S by any finite number of terms, each of which is smaller (w.r.t. $\succ$) than one of the replaced terms.

To combine these two concepts of tuples and multisets, we assign a status $\tau(f)$ to each operator $f \in \mathfrak{F}$ that determines the order according to which the subterms of f are compared. Formally, a status is a function which maps the set of operators into the set {mult , left , right}. Thus, a function symbol can have one of the following three values for its status:

- mult (the arguments will be compared as multisets),
- left (lexicographical comparison from left to right) and
- right (the arguments will lexicographically be compared from right to left).

The result of an application of the function args (cf. section 1.2) to a term $t = f(t_1, \ldots, t_n)$ depends on the status of f: If $\tau(f)$ = mult, then args(t) is the multiset $\{t_1, \ldots, t_n\}$. Otherwise, the tuple $(t_1, \ldots, t_n)$ can be obtained from args(t).

With the help of the concept of lexicographic concatenation of orderings (see 6.1) we are able to define an improved ordering on polynomial interpretations as follows:

Definition 6.2.1 (based on [La79])

Let [...] be a polynomial interpretation for variable terms that maps $\mathfrak{F}$ to $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$. Further, let $\triangleright$ be a precedence on $\mathfrak{F}$. If f is a unary operator interpreted as the identity function (i.e. $[f](x) = x$), there must not exist another operator being greater w.r.t. $\triangleright$.

The improved polynomial ordering $>_{IPOL}$ on terms s and t is defined as

$$s >_{IPOL} t \quad \text{iff} \quad \begin{array}{l} - \; s >_{POL} t \\ - \; top(s) \triangleright top(t) \\ - \; args(s) >_{IPOL,\tau(top(s))} args(t) \end{array}$$ ∎

The index $\tau(top(s))$ of $>_{IPOL,\tau(top(s))}$ marks the extension of $>_{IPOL}$ w.r.t. the status of the operator $top(s) =: f$:

$$\begin{array}{llll} (s_1,...,s_m) >_{IPOL,\tau(f)} (t_1,...,t_n) & & & \\ \text{iff} \quad \tau(f) = mult & \wedge & \{s_1,...,s_m\} \gg_{IPOL} \{t_1,...,t_n\} \\ \quad \text{or } \tau(f) = left & \wedge & (s_1,...,s_m) >^{lex}_{IPOL} (t_1,...,t_n) \\ \quad \text{or } \tau(f) = right & \wedge & (s_m,...,s_1) >^{lex}_{IPOL} (t_n,...,t_1) \end{array}$$

The equivalence of two terms is defined as $s =_{IPOL} t$ iff $s =_{POL} t \wedge top(s) = top(t) \wedge args(s) =_{IPOL,\tau(top(s))} args(t)$.

The idea of this ordering goes back to Lankford. His improvement of the original polynomial ordering is a simple version of $>_{IPOL}$ that assigns left-to-right status to all operators. It is influenced by the original Knuth-Bendix ordering (see chapter 7). Instead of using restricted functions as interpretations, however, general polynomials are permitted.

The following theorem summarizes the important properties of this new ordering.

Theorem 6.2.2

The improved polynomial ordering $>_{IPOL}$ is a simplification ordering on Γ and it is stable by instantiations. ∎

Note that $>_{IPOL}$ is a simplification ordering only if the condition "$f(x) >_{IPOL} x$ if $[f](x) = x$" is additionally required in definition 6.2.1 (analogous with the Knuth-Bendix ordering).

6.3 Examples

We conclude this chapter by demonstrating the practical applicability of the improved polynomial ordering as well as of the polynomial ordering extended to exponential functions. First of all, some examples are given which are readily obtained with $>_{IPOL}$ but not with the help of the original polynomial ordering.

Example 6.3.1 Division

$$
\mathfrak{R}: \quad \begin{array}{lcl} x / x & \rightarrow & 1 \\ x / 1 & \rightarrow & x \\ i(x / y) & \rightarrow & y / x \\ (x / y) / z & \rightarrow & x / (z / i(y)) \end{array}
$$

The interpretation for $>_{IPOL}$ guaranteeing the termination of $\mathfrak{R}$ could be of the form

$$
\begin{array}{lcl} [/](x, y) & = & x + y \\ [i](x) & = & x \\ [1]() & = & 2 \\ i \triangleright / \ , \ \tau(/) & = & \text{left} \end{array}
$$

The orientation of the first and the second equation is obvious. Both sides of the third rule are identical w.r.t. $>_{POL}$ and the top level symbol of the left-hand side is greater than that of the right-hand side ($i \triangleright /$). The interpretations of the terms of the last rule are identical, too. Furthermore, the leading operators are the same w.r.t. the precedence. This is why we have to compare the first arguments (due to the status of /), i.e. $x / y >_{IPOL} x$ since $x / y >_{POL} x$. ∎

Example 6.3.2 Summation

The following rule system defines the summation function on natural numbers, i.e. $f(n) = \sum_{i=0}^{n} i$.

$$
\mathfrak{R}: \quad \begin{array}{lcl} f(0) & \rightarrow & 0 \\ f(s(x)) & \rightarrow & s(x) + f(x) \\ x + 0 & \rightarrow & x \\ x + s(y) & \rightarrow & s(x + y) \end{array}
$$

By using $>_{IPOL}$ and the interpretations

$$
\begin{array}{lcl}
[f](x) & = & x^2 \\
[+](x, y) & = & x + y \\
[s](x) & = & x + 1 \\
[0]() & = & 2 \\
+ \triangleright s & &
\end{array}
$$

the termination of $\Re$ can be guaranteed. The only crucial rule is the last one (the other ones can already be oriented using $>_{POL}$):
$x + s(y) =_{POL} s(x + y) \quad \wedge \quad + \triangleright s.$ ∎

Example 6.3.3 ([KNZ86])

$$
\begin{array}{llcl}
\Re: & b + a & \rightarrow & a + b \\
& (x + y) + z & \rightarrow & x + (y + z) \\
& b + (a + z) & \rightarrow & a + (b + z) \\
& f(a, x) & \rightarrow & a \\
& f(b, x) & \rightarrow & b \\
& f(x + y, z) & \rightarrow & f(x, z) + f(y, z)
\end{array}
$$

The termination of $\Re$ can be guaranteed w.r.t. $>_{IPOL}$ based on the interpretations

$$
\begin{array}{lcl}
[+](x, y) & = & x + y \\
[f](x, y) & = & xy \\
[b]() & = & 3 \\
[a]() & = & 2 \\
f \triangleright + ,\ \tau(+) & = & \text{left}
\end{array}
$$

∎

Example 6.3.4 Addition & Multiplication

The definition of the multiplication operator on natural numbers (see example 6.1.1)

$$
\begin{array}{llcl}
\Re: & x * 0 & \rightarrow & 0 \\
& x * s(y) & \rightarrow & x + (x * y) \\
& x + 0 & \rightarrow & x \\
& x + s(y) & \rightarrow & s(x + y)
\end{array}
$$

can be oriented w.r.t. $>_{IPOL}$ using the following polynomial interpretations:

$$
\begin{array}{lcl}
[*](x, y) & = & xy \\
[+](x, y) & = & x + y \\
[s](x) & = & x + 1
\end{array}
$$

$[0]() = 2$

$* \triangleright + \triangleright s$ ■

These examples demonstrate that $>_{IPOL}$ is a proper improvement of $>_{POL}$, i.e. the set of all rule systems which can be oriented w.r.t. $>_{POL}$ is properly included in the corresponding set of $>_{IPOL}$. In addition, the improved polynomial ordering is also more readily obtained than the original one. The following example illustrates this assertion.

Example 6.3.5 Reverse operation on lists

$\mathfrak{R}$:	nil ∘ y	→	y
	(x.y) ∘ z	→	x.(y ∘ z)
	rev(nil)	→	nil
	rev(x.y)	→	rev(y) ∘ (x.nil)
	rev(rev(x))	→	x

The original polynomial ordering orient $\mathfrak{R}$, but it is difficult to find appropriate interpretations: $[\circ](x,y) = xy$, $[.](x,y) = xy + y + 1$, $[rev](x) = x^2$ and $[nil]() = 2$. On the contrary, it is relatively simple to deduce the following interpretations for $>_{IPOL}$:

$[\circ](x,y) = x + y$

$[.](x,y) = x + y$

$[rev](x) = x^2$

$[nil]() = 2$

$\circ \triangleright .$ ■

Let us consider some rule systems which cannot be oriented w.r.t. $>_{POL}$. By using more complex functions than polynomials their termination can be proved. Obviously, it is possible to extend $>_{IPOL}$ by replacing polynomials by other functions.

Example 6.3.6 Summation (see 6.3.2)

$\mathfrak{R}$:	f(0)	→	0
	f(s(x))	→	s(x) + f(x)
	x + 0	→	x
	x + s(y)	→	s(x + y)

The following interpretations suffice to prove the termination of $\mathfrak{R}$:

$$
\begin{aligned}
[+](x,y) &= x + 2y\\
[f](x) &= 2^{x}\\
[s](x) &= x + 2\\
[0](\,) &= 2
\end{aligned}
$$

Note that $\mathfrak{R}$ can also be oriented with the help of $>_{IPOL}$ (cf. 6.3.2). ∎

Example 6.3.7 Disjunctive normal form ([De83])

The rule system

$$
\mathfrak{R}:\quad
\begin{aligned}
\neg(x \wedge y) &\rightarrow \neg x \vee \neg y\\
\neg(x \vee y) &\rightarrow \neg x \wedge \neg y\\
x \wedge (y \vee z) &\rightarrow (x \wedge y) \vee (x \wedge z)
\end{aligned}
$$

denoting the computation of disjunctive normal forms requires a more complex interpretation than a polynomial one:

$$
\begin{aligned}
[\wedge](x,y) &= xy\\
[\vee](x,y) &= x + y + 1\\
[\neg](x) &= 2^{x}
\end{aligned}
$$

∎

The following examples describe functions which values increase quicker than polynomials. Intuitively, the increase of the values of the interpretations used to prove the termination has to correspond to that of the functions.

Example 6.3.8 Exponential function

$$
\mathfrak{R}:\quad
\begin{aligned}
x^{0} &\rightarrow s(0)\\
x^{s(y)} &\rightarrow x * x^{y}\\
0 * y &\rightarrow 0\\
s(x) * y &\rightarrow y + (x * y)
\end{aligned}
$$

An appropriate interpretation is of the form

$$
\begin{aligned}
[exp](x,y) &= 2^{xy}\\
[*](x,y) &= xy\\
[+](x,y) &= x + y\\
[s](x) &= 2x\\
[0](\,) &= 2
\end{aligned}
$$

∎

Example 6.3.9 Factorial function

$$
\begin{array}{llll}
\mathfrak{R}: & f(0) & \rightarrow & s(0) \\
& f(s(x)) & \rightarrow & s(x) * f(x) \\
& x * 0 & \rightarrow & 0 \\
& x * s(y) & \rightarrow & (x * y) + x \\
& x + 0 & \rightarrow & 0 \\
& x + s(y) & \rightarrow & s(x + y)
\end{array}
$$

We choose the interpretations

$$
\begin{array}{lll}
[*](x, y) & = & xy \\
[+](x, y) & = & x + 2y \\
[f](x) & = & 2x^2 \\
[s](x) & = & x + 3 \\
[0]() & = & 2
\end{array}
$$

■

Example 6.3.10 Fibonacci function

The rule system

$$
\begin{array}{llll}
\mathfrak{R}: & f(0) & \rightarrow & s(0) \\
& f(s(0)) & \rightarrow & s(0) \\
& f(s(s(x))) & \rightarrow & f(s(x)) + f(x) \\
& x + 0 & \rightarrow & x \\
& x + s(y) & \rightarrow & s(x + y)
\end{array}
$$

specifying the well-known Fibonacci function is terminating since the ordering on the following interpretations can orient its rules in the desired way:

$$
\begin{array}{lll}
[+](x, y) & = & x + 2y \\
[f](x) & = & 2^{x+1} \\
[s](x) & = & x + 2 \\
[0]() & = & 1
\end{array}
$$

■

We were able to prove the termination of some examples using a generalized polynomial ordering by applying exponential functions. The last example (6.3.11) of this section illustrates the limits of this technique. Note that the division specification of 6.3.1 cannot be oriented using exponential functions, either. However, its termination can be proved with the help of $>_{IPOL}$.

Example 6.3.11 Ackermann function

It is known that the specification of the Ackermann function terminates. Unfortunately, we are not able to prove this property with any of the techniques presented in this paper.

$$\mathfrak{R}: \begin{array}{lcl} A(0,y) & \to & s(y) \\ A(s(x),0) & \to & A(x,s(0)) \\ A(s(x),s(y)) & \to & A(x,A(s(x),y)) \end{array}$$

Note that the recursive path ordering with status (see section 7.1) can orient $\mathfrak{R}$ (if it is based on $A \triangleright s$, $\tau(A) = left$). ∎

6.4 Discussion

In this chapter, we have presented a polynomial ordering based on the lexicographic concatenation of well-founded orderings. More precisely, the new ordering $>_{IPOL}$ (based on an approach of Lankford, [La79]) is defined as the concatenation of the original polynomial ordering $>_{POL}$, a precedence and the comparison of the arguments w.r.t. the recursive definition of the whole ordering which additionally considers a status function (see [St89a]). This new ordering is a simplification ordering possessing the subterm property provided the polynomials that are used to interpret the function symbols are simplifying. In particular, it is not allowed to use identity functions as interpretations of unary operators which are not maximal w.r.t. the precedence (cf. Knuth-Bendix ordering), e.g. with $[g](x) = x \wedge f \triangleright g$ we could orient the rule $f(x) \rightarrow g(f(x))$ which does not terminate in any case. All other interpretations are simplifying since we use the definition of chapter 2.

There also exists another similar extension of the original polynomial ordering which is known as the Cartesian product of polynomials. Note that example 6.3.1 (the division of natural numbers) cannot be oriented w.r.t. this ordering. However, $>_{IPOL}$ is able to prove its termination. On the contrary, the example 2.4.12 cannot be oriented w.r.t. $>_{IPOL}$ whereas it is possible with the help of the Cartesian product of polynomials (using $>_1 = >_{POL}$ based on $[*](x,y) = xy$, $[+](x,y) = 2x + y + 1$, $[0]() = 9$, $[1]() = 8$, $[2]() = 2$, $[a]() = 2$ and $>_2 = >_{POL}$ based on $[+](x,y) = xy + x$, $[0]() = 2$, $[1]() = 3$).

Note that $>_{IPOL}$ can be further extended by using functions which complexities are higher than those of polynomials. We are of the opinion that this combination is intrinsically more powerful than all other known versions of polynomial orderings (see 6.3.6 - 6.3.10).

During the computation of a lot of examples (see [SK90]) we had the interesting experience that the improved polynomial ordering proved to be easier to handle than the original one since it is relatively easy to make two equivalent (w.r.t. a theory) terms equivalent w.r.t $=_{POL}$ by using the semantics (if known) of the operators. For example, consider the definition of the addition on natural numbers: $x + s(y) \rightarrow s(x + y)$. The semantic interpretations $[+](x,y) = x + y$ and $[s](x) = x + 1$ lead to the equality of the polynomials $(x + y + 1)$. Adding $+ \triangleright s$ to the precedence the termination of the rule is proved.

It is obvious that there exists a close relationship between $>_{IPOL}$ and the polynomial ordering modulo theories (see chapter 5) since the equivalence w.r.t. $=_{POL}$ is part of $>_{IPOL}$ (see above). Therefore, the results obtained for polynomial orderings that are compatible w.r.t. underlying theories can help to find an appropriate $>_{IPOL}$. This is one more reason for the improved polynomial ordering being more tractable than the original polynomial ordering.

7 Comparison with Other Orderings

From a theoretical point of view polynomial orderings provide a powerful technique for proving the termination of term rewriting systems. To illustrate their power we relate them to other orderings. In this chapter, we compare various kinds of polynomial orderings with path and decomposition orderings as well as with the Knuth-Bendix ordering. Furthermore, a comparison of polynomial orderings restricted to underlying AC-theories with other AC-orderings is enclosed.

7.1 Definitions of Orderings

Chapter 6 contains a comparison of the various versions of polynomial orderings. In the present chapter we extend this comparison by including other classes of orderings: the path and decomposition orderings and the Knuth-Bendix ordering.

The class of path and decomposition orderings is very extensive. For compactness, we will only give a rough description of their essential characteristics. Path orderings compare two terms by comparing all their paths w.r.t. a multiset ordering. A path of a term is a sequence of terms starting with the whole term followed by a path of one of its arguments. The most well-known path orderings are the path of subterms ordering of Plaisted ([Pl78]) and the path ordering of Kapur, Narendran and Sivakumar ([KNS85]). Decomposition orderings are similar to path orderings. They differ from path orderings by comparing decompositions instead of paths. However, decompositions are paths for which the order of succession on terms is irrelevant. Thus, a decomposition represents a set (instead of a sequence) of terms. The first decomposition ordering is called recursive decomposition ordering and has been developed by Lescanne, Jouannaud and Reinig (see for example [JLR82]). For more information on decomposition orderings as well as path orderings and their comparison, see [De87] and [St89a].

A common origin of all path and decomposition orderings is the wide-spread recursive path ordering by Dershowitz (cf. [De82]). We have chosen this technique as a representative for the path and decomposition orderings since it combines the features of the orderings of all these classes. The following definition is the one of Kamin and Lévy who extended the original recursive path ordering by incorporating a status function (see [KL80]).

Definition 7.1.1 ([KL80], [De82])

Let $\triangleright$ be a precedence on $\mathfrak{F}$. The recursive path ordering with status $>_{RPOS}$ on terms s and t is defined as

$s >_{RPOS} t$

iff i) $top(s) \triangleright top(t) \wedge \{s\} \gg_{RPOS} args(t)$

ii) $top(s) = top(t) \wedge \tau(top(s)) = mult \wedge args(s) \gg_{RPOS} args(t)$

iii) $top(s) = top(t) \wedge \tau(top(s)) \neq mult \wedge \{s\} \gg_{RPOS} args(t)$
$\wedge\ args(s) >_{RPOS,\tau(top(s))} args(t)$

iv) $args(s) \underline{\gg}_{RPOS} \{t\}$ ■

The method of comparing two terms w.r.t. the recursive path ordering with status (RPOS) depends on the leading function symbols. The relationship between these two operators w.r.t. the precedence $\triangleright$ is responsible for decreasing one (or both) of the terms in the recursive definition of the RPOS. If one of the terms is 'empty' (i.e. totally decreased) then the other one is greater.

Example 7.1.2 Ring

The rule system

$$
\mathfrak{R}:\quad \begin{array}{lcl}
x + 0 & \rightarrow & x \\
x + i(x) & \rightarrow & 0 \\
(x + y) + z & \rightarrow & x + (y + z) \\
i(x + y) & \rightarrow & i(y) + i(x) \\
x * (y + z) & \rightarrow & (x * y) + (x * z) \\
(x + y) * z & \rightarrow & (x * z) + (y * z)
\end{array}
$$

which can be completed to a specification of a ring is terminating since the recursive path ordering based on the precedence

$$* \triangleright i \triangleright + \triangleright 0$$

and on the status function

$$\tau(*) = \tau(+) = \text{left}$$

orients the rules of $\mathfrak{R}$ in the desired direction. Note that $\mathfrak{R}$ cannot be oriented with the help of any polynomial ordering. ■

Another well-known ordering (which is of historical importance) is the one of Knuth and Bendix (see [KB70]). We will present an improvement of their technique which permits the use of a status function (see [St89a]). Like the original Knuth-Bendix ordering, the Knuth-Bendix ordering with status (KBOS, for short) assigns natural (or possibly real) numbers to the function symbols and then to terms (called weight of a term) by adding the numbers of the operators they contain. Two terms are compared by comparing their weights. If their weights are equal the subterms are lexicographically collated. In order to define this technique we need some helpful denotations.

If x is a variable and t is a term we denote the number of occurrences of x in t by $\#_x(t)$. We assign a non-negative integer $\varphi(f)$ - the weight of f - to each operator in $\mathfrak{F}$ and a positive integer φ_0 to each variable such that

$$
\begin{array}{ll}
\varphi(c) \geq \varphi_0 & \text{if } c \text{ is a constant and} \\
\varphi(f) > 0 & \text{if } f \text{ has one argument.}
\end{array}
$$

Furthermore, the weight of a term is defined as $\varphi(t) = \varphi(f) + \Sigma\varphi(t_i)$ if $t = f(t_1,\ldots,t_n)$.

Definition 7.1.3 ([KB70], [St89a])

Let $\triangleright$ be a precendence and φ a weight function (as described above). The Knuth-Bendix ordering $>_{KBOS}$ with status on terms s and t is defined as

$$s >_{KBOS} t$$
$$\text{iff} \quad (\forall x \in \mathfrak{X})\ \#_x(s) \geq \#_x(t) \quad \wedge$$
$$- \quad \varphi(s) > \varphi(t)$$
$$- \quad top(s) \triangleright top(t)$$
$$- \quad args(s) >_{KBOS,\tau(top(s))} args(t)$$ ∎

This ordering is closely related to the improved polynomial ordering (see definition 6.2.1) since it has the same structure. The difference between the Knuth-Bendix ordering with status and $>_{IPOL}$ is the fact that the Knuth-Bendix ordering uses only super-linear interpretations.

Lemma 7.1.4

Let $>_{IPOL}$ be based on super-linear interpretations, only. Then,

$$s >_{KBOS} t \quad \text{iff} \quad s >_{IPOL} t$$ ∎

The requirement of super-linear interpretations induces the so-called variable condition of the Knuth-Bendix ordering (with status): The multiset of variables of the right-hand side of a rule must be contained in that of the left-hand side. This condition guarantees the stability w.r.t. instantiations. However, it is a very strong restriction. Note that, for example, the distributivity law cannot be oriented in the usual direction.

Remark 7.1.5

Analogous with polynomial orderings the weight of an operator may sometimes be zero (this assignment corresponds to the identity polynomial). Note that the weight must be greater than zero if the operator represents a constant symbol.
Another exception requires that a unary operator with weight zero must be a maximum w.r.t. the precedence (cf. definition 6.2.1). ∎

Example 7.1.6

Let $\mathfrak{R}$: $f(f(x)) \rightarrow g(g(x))$
$g(g(f(x))) \rightarrow f(g(g(x)))$

The Knuth-Bendix ordering based on the weight function

$$\varphi(f) = 2$$
$$\varphi(g) = 1$$

(which corresponds to the super-linear interpretations $[f](x) = x + 2$, $[g](x) = x + 1$) can orient $\mathfrak{R}$ if the precedence contains the relation $g \triangleright f$. Note that the termination of $\mathfrak{R}$ is also guaranteed with the help of the original polynomial ordering using $[f](x) = 2x + 1$, $[g](x) = 2x$. It is obvious that $\mathfrak{R}$ cannot be oriented w.r.t. any recursive path ordering (in particular, this is not possible with any path or decomposition ordering, either). ∎

7.2 Comparison

In this section we compare the power of the following orderings:

- $>_{PATH}$ which stands for any technique belonging to the class of path and decomposition orderings (e.g. $>_{RPOS}$)
- $>_{KBOS}$
- $>_{POL}$
- $>_{IPOL}$
- $>_{EXP}$ which represents the method of using exponential functions as interpretations in addition to polynomials

Note that these orderings relate to three parameters: the interpretation, the precedence and the status function. Therefore, we use $>(i, p, \tau)$ to denote the ordering $>$ which is based on the interpretation i, the precedence p and the status τ.

The power of an ordering is represented by the set of comparable terms. We will examine the relation between two sets. The following three relations will be examined.

Definition 7.2.1

Let $>$ and $\succ$ be two orderings. Further, let i, i', i'' be interpretations, p, p', p'' precedences and τ, τ', τ'' status functions.

- Two orderings are equivalent:

$$> \; = \; \succ \quad \text{iff} \quad s >(i, p, \tau) \; t \;\leftrightsquigarrow\; s \succ(i, p, \tau) \; t$$

- One ordering is properly included in the other one:

$$\begin{aligned} > \; \subset \; \succ \quad \text{iff} \quad & (\exists i', p', \tau')\, s >(i, p, \tau) \; t \;\rightsquigarrow\; s \succ(i', p', \tau') \; t \\ \wedge \quad & (\exists i', p', \tau')(\nexists i'', p'', \tau'') \; s \succ(i', p', \tau') \; t \;\wedge\; s >(i'', p'', \tau'') \; t \end{aligned}$$

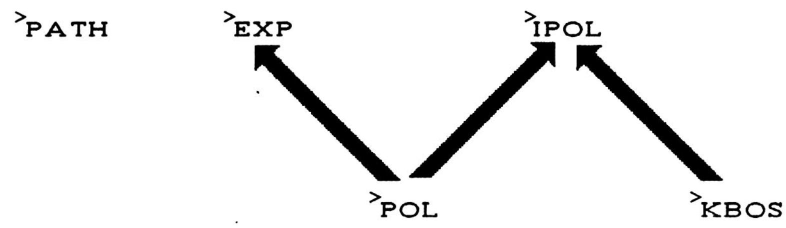
- Two orderings overlap with each other:

$$\begin{aligned} > \; \# \; \succ \quad \text{iff} \quad & (\exists i', p', \tau')(\nexists i'', p'', \tau'') \; s >(i', p', \tau') \; t \;\wedge\; s \succ(i'', p'', \tau'') \; t \\ \wedge \quad & (\exists i', p', \tau')(\nexists i'', p'', \tau'') \; s \succ(i', p', \tau') \; t \;\wedge\; s >(i'', p'', \tau'') \; t \end{aligned}$$

■

The following lemma reflects the results obtained by the comparison of the orderings described in this report. To present the relations in a simple way, we use a kind of Hasse diagrams. If $> \subset \succ$, then we arrange $\succ$ above $>$ joining them with an arrow.

Lemma 7.2.2



■

Note that it is possible to combine $>_{EXP}$ and $>_{IPOL}$ (denoted by $>_{IPOL*}$) by only using $>_{IPOL}$ based on polynomials *and* exponential functions. This would change lemma 7.2.2 such that only the two orderings $>_{PATH}$ and $>_{IPOL*}$ instead of all three must be considered since they are more powerful than all the other orderings of the diagram. Furthermore, in section 6.1 we have illustrated another improvement of the original polynomial ordering: the Cartesian product of polynomials. It is obvious that $>_{IPOL*}$ can be further extended by applying this general technique of lexicographic concatenation of orderings to $>_{IPOL*}$. It stands to reason that the resulting ordering is more powerful than the original Cartesian product of polynomials.

Chapter 5 deals with special polynomial orderings which can be used to prove the termination of rewrite systems based on underlying theories. However, there are other orderings that are also compatible with a particular theory: the associative-commutative theory. The set of those orderings contains the associative path ordering $>_{APO}$ (see for example [GL86]), the path and decomposition orderings for AC-theories ([St89d] and [St90b]) and the associative-commutative Knuth-Bendix ordering $>_{ACK}$ ([St89b] and [St90a]). The following lemma describes the comparison of these orderings (we use $>_{APO}$ as a representative for the path and decomposition orderings modulo AC-theories) including the original polynomial ordering on AC-theories (denoted by $>_{ACP}$, see section 5.3).

Lemma 7.2.3

The orderings $>_{APO}$, $>_{ACK}$ and $>_{ACP}$ overlap with each other. ■

8 Conclusion

This chapter contains a resumé of the entire report. The known and the new results are specified separately.

Termination is a substantial property of ordinary term rewriting systems as well as of term rewriting systems with underlying theories. The report in hand gives a detailed account on a well-known technique proving this characteristic: the ordering on polynomial interpretations. This technique is a combination of the approaches of Knuth and Bendix (Knuth-Bendix ordering, [KB70]) and of Manna and Ness (a method for guaranteeing the termination of Markov-like algorithms, [MN70]). It generalizes both methods and was developed by Lankford ([La75], [La76] and [La79]). Polynomial orderings are based on the idea of assigning polynomials to function symbols in order to define a mapping from terms to polynomials. Two terms are compared by comparing their corresponding polynomials. As a result, the following problems are to be encountered:

- Procedures for generating adequate interpretations for operators
- Methods for deciding whether a polynomial is greater than zero

It should be noticed that the second problem is undecidable for polynomials over $\mathbb{N}$ ([Da73]).

We have dealt with well-known in addition to new techniques for solving these tasks. Furthermore, we have discussed some known modifications of the original polynomial ordering. We did our best endeavours to illustrate the explanations by various examples.

The following well-known facts have been presented:

- We have dealt with strategies for the generation of adequate interpretations which guarantee the termination of a given rule system. Some heuristics of Ben Cherifa ([Be86]) have been considered including suggestions for orienting associativity laws and homeomorphism rules (see 3.1).

- Sufficient solutions for the problem of deciding whether a polynomial is greater than zero are discussed in chapter 4. Ben Cherifa and Lescanne developed a procedure for implementing the theoretical ideas of [La79] and [HO80]. Starting from the initial problem $p_0 \overset{?}{>} 0$, they build a sequence of inequalities such that $p_0 \geq p_1 \geq \ldots \geq p_n > 0$. The positiveness of p_n is supposed to be checked by a basic principle like 'all coefficients are positive'. At each step they set off a monomial with a negative coefficient against a convenient positive monomial. A more powerful approach is that of Rouyer ([Ro88]). The motivation is that a polynomial which does not have any root w.r.t. an interval $[a, \infty]$ is positive w.r.t. $[a, \infty]$ if the leading coefficient is positive. With the help of the Sturm sequence (see [Du60]), it is possible to compute the number of roots w.r.t. $[a, \infty]$.

- The use of rewriting systems which are based on an underlying theory E presumes E-termination. If the polynomial ordering is E-compatible

($s =_E t \rightsquigarrow [s] =_{POL} [t]$) it is suited for proving this property. In order to fulfil the compatibility, the classes of polynomials must be restricted. In [Be86], these restrictions are fixed for the case that E is associative-commutative-distributive, permutative, transitive or idempotent (see chapter 5).

- An examination of different extensions of the original polynomial ordering is discussed in chapter 6. One possible improvement concerns the functions used as interpretations. For example, it is acceptable to include exponential functions while observing the main conditions. Furthermore, the concept of lexicographic concatenation of orderings can be applied to the polynomial ordering in such a way that several different polynomial orderings (based on different interpretations) are concatenated. This technique is called the Cartesian product of polynomials. A third improvement of a polynomial ordering which is similar to the last point involves the concatenation of the original polynomial ordering, a precedence and the comparison of the arguments (in lexicographical order) w.r.t. the recursive definition of the entire ordering (see [La79]).

In addition to the discussion of the above aspects of polynomial orderings, we have obtained some new results:

- From a practical point of view it is interesting to simplify the polynomials used as interpretations such that it will be easier to check whether a polynomial is greater than zero. Our implementation is based on the idea of using only polynomials which are monotonous and non-diminishing (see chapter 2).
- A series of 300 experiments has been conducted to better help us in our choice of the interpretations (cf. [SK90]). This examination leads to new heuristics about arithmetic theories, distributivity axioms and group theories (see chapter 3). A criterion that can sometimes detect the necessity of mixed interpretations is also available. The termination proofs of many of the examples checked only need linear interpretations. Therefore, we have developed an algorithm that automatically generates linear interpretations for a given rule system. The constitutive concept is the transformation of inequalities of nearly general polynomials into inequalities of linear polynomials by an approximation. An application of the known Simplex method (see 3.4) eventually computes a solution of this system of linear inequalities.
- The field of polynomial orderings for underlying theories has been extended by specifying restricted interpretations for special commutativity,

distributivity, endomorphism, minus and unipotency. Moreover, special theories like abelian groups, quasi-groups, boolean rings and if-then-else axioms have been treated (see chapter 5).

- Chapter 6 deals with an improvement of the extended polynomial ordering of Lankford (see above). We have incorporated a status function as was possible with the Knuth-Bendix ordering (see [St89a]).
- A conclusive comparison of the various kinds of polynomial orderings presented including path and decomposition orderings as well as the Knuth-Bendix ordering polishes our report (see chapter 7). One of the most interesting results of this confrontation concerns the fact that the path and decomposition orderings overlap with all classes of polynomial orderings.

As we have mentioned, polynomial orderings are a powerful tool for proving the (E-) termination of rewrite systems. They enable the integration of the semantics of the used operators into the termination proof. Notably, a term rewriting system whose termination has been proven by orderings based on polynomial interpretations can even have derivations of double exponential length (cf. [La88], [HL89] and [Ho90]). A negative aspect are the difficulties we usually faces in practical applications and which make path and decomposition orderings appear a lot more advantageous. Thus, the minimization of this deficiency should be part of future plans.

Acknowledgement

There remains the pleasant task of thanking those who have co-operated in the production of this paper: Jürgen Avenhaus, Christopher Bewick, Roland Fettig, Rita Kohl, Ulrich Kühler, Klaus Madlener and Inger Sonntag.

References

[AGGMS87] Jürgen Avenhaus, Richard Göbel, Bernhard Gramlich, Klaus Madlener, Joachim Steinbach
TRSPEC: A term rewriting based system for algebraic specifications
Proceedings of the 1st International Workshop on Conditional Term Rewriting Systems, Orsay (Paris), July 1987, LNCS 308, pp. 245-248

[AM89] Jürgen Avenhaus, Klaus Madlener
Term rewriting and equational reasoning
Formal Techniques in Artificial Intelligence: A sourcebook, R.B. Banerji (ed.), Elsevier Science Publishers B.V., Amsterdam, 1989

[AMS89] Jürgen Avenhaus, Klaus Madlener, Joachim Steinbach
COMTES - An experimental environment for the completion of term rewriting systems
Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill, U.S.A., April 1989, LNCS 355, pp. 542-546

[Av89] Jürgen Avenhaus
Reduktionssysteme
Skript zu einer Vorlesung im Sommersemester 1989, Fachbereich Informatik, Universität Kaiserslautern, 1989

[BB61] Edwin Beckenbach, Richard Bellman
An introduction to inequalities
The Mathematical Association of America, New mathematical library, Twelfth Printing, 1961

[Be86] Ahlem Ben Cherifa
Preuves de terminaison des systèmes de réécriture - un outil fondé sur les interprétations polynomiales
Thèse de doctorat de l'université de Nancy I en informatique, Octobre 1986

[BL85] Ahlem Ben Cherifa, Pierre Lescanne
A method for proving termination of rewriting systems, based on elementary computations on polynomials
Proceedings of the Workshop on Combinatorial Algorithms in Algebraic Structures, Otzenhausen, West Germany, 1985, pp. 6-13

[BL86] Ahlem Ben Cherifa, Pierre Lescanne
An actual implementation of a procedure that mechanically proves termination of rewriting systems based on inequalities between polynomial interpretations
Proceedings of the 8th International Conference on Automated Deduction, Oxford, England, 1986, LNCS 230, pp. 42-51

[BL87] Ahlem Ben Cherifa, Pierre Lescanne
Termination of rewriting systems by polynomial interpretations and its implementation
Science of Computer Programming 9 (2), October 1987, pp. 137-160, also Internal Report, Centre de Recherche en Informatique de Nancy, Nancy, France

[BP85] Leo Bachmair, David A. Plaisted
Termination orderings for associative-commutative rewriting systems
Journal of Symbolic Computation 1, 1985, pp. 329-349

[Ch83] Vašek Chvátal
Linear Programming
W.H. Freeman and Company, New York/San Francisco, ISBN 0-7167-1195-8, 1983

[Ch89] Jim Christian
Fast Knuth-Bendix completion: Summary
Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill, North Carolina, U.S.A., April 1989, LNCS 355, pp. 551-555

[Co69] Paul J. Cohen
Decision procedures for real and p-adic fields
Communications on Pure and Applied Mathematics, Vol. 22, 1969, pp. 131-151

[Co75] George E. Collins
Quantifier elimination for real closed fields by cylindrical algebraic decomposition
Proceedings of the 2nd GI Conference on Automata and Formal Languages, Springer Verlag, 1975, LNCS 33, pp. 134-183

[Da73] Martin Davis
Hilbert's tenth problem is unsolvable
Amer. Math. Monthly 80 (3), March 1973, pp. 233-269

[Da88] Max Dauchet
Termination of rewriting is undecidable in the one-rule case
Proceedings of the 13th Symposium of Mathematical Foundations of Computer Science, Carlsbad, CSSR, September 1988, LNCS 324, pp. 262-270

[De79] Nachum Dershowitz
A note on simplification orderings
Information Processing Letters 9 (5), November 1979, pp. 212-215

[De82] Nachum Dershowitz
Orderings for term rewriting systems
Journal of Theoretical Computer Science 17 (3), March 1982, pp. 279-301

[De83] Nachum Dershowitz
Well-founded orderings
Technical Report ATR-83(8478)-3, Information Sciences Research Office, The Aerospace Corporation, El Segundo, California, May 1983

[De87] Nachum Dershowitz
Termination of rewriting
Journal of Symbolic Computation 3, 1987, pp. 69-116

[DHJP83] Nachum Dershowitz, Jieh Hsiang, N. Alan Josephson, David A. Plaisted
Associative-commutative rewriting
Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, August 1983, pp. 940-944

[DM79] Nachum Dershowitz, Zohar Manna
Proving termination with multiset orderings
Communications of the Association for Computing Machinery, Vol. 22, No. 8, August 1979, pp. 465-476

[Du60] Emile Durand
Solutions numériques des équations algébriques
Masson, 1960

[GL86] Isabelle Gnaedig, Pierre Lescanne
Proving termination of associative-commutative rewriting systems by rewriting
Proceedings of the 8th Conference on Automated Deduction, Oxford, England, 1986, LNCS 230, pp. 52-60

[Gn88] Isabelle Gnaedig
Total orderings for equational theories
Working document, Nancy, France, 1988

[HL78] Gérard Huet, Dallas S. Lankford
On the uniform halting problem for term rewriting systems
Rapport Laboria 283, IRIA, Paris, Institut de Recherche d'Informatique et d'Automatique Rocquencourt, France, March 1978

[HL89] Dieter Hofbauer, Clemens Lautemann
Termination proofs and the length of derivations
Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill, North Carolina, U.S.A., April 1989, LNCS 355, pp. 167-177

[HO80] Gérard Huet, Derek C. Oppen
Equations and rewrite rules: A survey
in: Formal languages - Perspectives and open problems, R. Book (ed.), Academic Press, New York, 1980, pp. 349-405

[Ho90] Dieter Hofbauer
Terminationsbeweise und Ableitungslängen
Proceedings 2. Workshop "Termersetzung: Grundlagen und Anwendung", Dortmund, BRD, März 1990, Forschungsbericht Nr. 336, FB Informatik, Universität Dortmund, p. 23

[Hu80a] Jean-Marie Hullot
A catalogue of canonical term rewriting systems
Technical Report CSL-113, SRI International, Menlo Park, California, April 1980

[Hu80b] Gérard Huet
Confluent reductions: Abstract properties and applications to term rewriting systems
Journal of the Association for Computing Machinery, Vol. 27, No. 4, October 1980, pp. 797-821

[JK86] Jean-Pierre Jouannaud, Hélène Kirchner
Completion of a set of rules modulo a set of equations
Journal on Computing 15 (4) of the Society for Industrial and Applied Mathematics, November 1986, pp. 1155-1194

[JL87] Jean-Pierre Jouannaud, Pierre Lescanne
Rewriting systems
Technology and Science of Informatics 6 (3), 1987, pp. 181-199

[JLR82] Jean-Pierre Jouannaud, Pierre Lescanne, Fernand Reinig
Recursive decomposition ordering
Working Conference on Formal Description of Programming Concepts II of the International Federation for Information Processing, D. Bjørner (ed.), North Holland, Garmisch Partenkirchen, W. Germany, 1982, pp. 331-348

[JM84] Jean-Pierre Jouannaud, Miguel Muñoz
Termination of a set of rules modulo a set of equations
Proceedings of the 7th Conference on Automated Deduction, Napa, California, 1984, LNCS 170, pp. 175-193

[Jo83] Jean-Pierre Jouannaud
Confluent and coherent equational term rewriting systems - Applications to proofs in abstract data types
Proceedings of the International Conference on Automata, Algebra and Programming, L'Aquila, Italy, 1983, LNCS 159, pp. 269-283

[KB70] Donald E. Knuth, Peter B. Bendix
Simple word problems in universal algebras
Computational problems in abstract algebra, J. Leech (ed.), Pergamon Press, 1970, pp. 263-297

[KL80] Sam Kamin, Jean-Jacques Lévy
Attempts for generalizing the recursive path orderings
Unpublished manuscript, Department of Computer Science, University of Illinois, Urbana, Illinois, U.S.A., February 1980

[KNS85] Deepak Kapur, Paliath Narendran, G. Sivakumar
A path ordering for proving termination of term rewriting systems
Proceedings of the 10th Colloquium on Trees in Algebra and Programming, Berlin, West Germany, March 1985, LNCS 185, pp. 173-187

[KNZ86] Deepak Kapur, Paliath Narendran, Hantao Zhang
Proof by induction using test sets
Proceedings of the 8th Conference on Automated Deduction, Oxford, England, July 1986, LNCS 230, pp. 99-117

[KS83] Deepak Kapur, G. Sivakumar
Experiments with and architecture of RRL, a rewrite rule laboratory
Proceedings of an NSF Workshop on the rewrite rule laboratory, Schenectady, New York, U.S.A., 1983, pp. 33-56

[Kü90] Ulrich Kühler
Erfahrungen mit dem rewrite-basierten Induktionsbeweiser TRSPEC
Project Report, Department of Computer Science, University of Kaiserslautern, West Germany, February 1990

[La75] Dallas S. Lankford
Canonical algebraic simplification in computational logic
Memo ATP-25, Automatic Theorem Proving Project, Department of Mathematics and Computer Science, University of Texas, Austin, Texas

[La76] Dallas S. Lankford
A finite termination algorithm
Internal Report, Department of Mathematics, Southwestern University, Georgetown, Texas, U.S.A., March 1976

[La79] Dallas S. Lankford
On proving term rewriting systems are noetherian
Memo MTP-3, Mathematics Department, Louisiana Technical University, Ruston, Louisiana, U.S.A., May 1979

[La86] Dallas S. Lankford
Some remarks on rewrite rule termination methods which use polynomial interpretations
Unpublished Abstract

[La88] Clemens Lautemann
A note on polynomial interpretation
Bulletin of the European Association for Theoretical Computer Science 36, October 1988, pp. 129-131

[LB77a] Dallas S. Lankford, A.M. Ballantyne
Decision procedures for simple equational theories with commutative axioms: Complete sets of commutative reductions
Technical Report ATP-35, Automatic Theorem Proving Project, Department of Mathematics and Computer Science, University of Texas, Austin, Texas, March 1977

[LB77b] Dallas S. Lankford, A.M. Ballantyne
Decision procedures for simple equational theories with permutative axioms: Complete sets of permutative reductions
Technical Report ATP-37, Automatic Theorem Proving Project, Department of Mathematics and Computer Science, University of Texas, Austin, Texas, April 1977

[LB77c] Dallas S. Lankford, A.M. Ballantyne
Decision procedures for simple equational theories with commutative-associative axioms: Complete sets of commutative-associative reductions
Technical Report ATP-39, Automatic Theorem Proving Project, Department of Mathematics and Computer Science, University of Texas, Austin, Texas, August 1977

[Le86] Pierre Lescanne
Divergence of the Knuth-Bendix completion procedure and termination orderings
Bulletin of the European Association for Theoretical Computer Science 30, 1986, pp. 80-83

[LS77] R.J. Lipton, L. Snyder
On the halting of tree replacement systems
Proceedings of a Conference on Theoretical Computer Science, Waterloo, Ontario, Canada, August 1977, pp. 43-46

[Ma87] Ursula Martin
How to choose the weights in the Knuth-Bendix ordering
Proceedings of the 2nd International Conference on Rewriting Techniques and Applications, Bordeaux, France, May 25-27, 1987, LNCS 256, pp. 42-53

[Mi76] Gautam Mitra
Theory and application of mathematical programming
Academic Press Inc. (London) Ltd., 1976

[MN70] Zohar Manna, Stephen Ness
On the termination of Markov algorithms
Proceedings of the third Hawaii International Conference on System Science, Honolulu, Hawaii, January 1970, pp. 789-792

[Mu80] David R. Musser
On proving inductive properties of abstract data types
Proceedings of the 7th Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada, U.S.A., 1980, pp. 154-162

[Pa89] Doerthe Paul
Implementierung von Polynomordnungen
Project report, Department of Computer Science, University of Kaiserslautern, West Germany, 1989

[Pl78] David A. Plaisted
A recursively defined ordering for proving termination of term rewriting systems
Internal Report, University of Illinois, Urbana, Illinois, U.S.A., September 1978

[PS81] Gerald Peterson, Mark E. Stickel
Complete sets of reductions for some equational theories
Journal of the Association of Computing Machinery 28 (2), April 1981, pp. 233-264

[Ro88] Jocelyne Rouyer
Preuves de terminaison de systèmes de réécriture fondées sur les interprétations polynomiales - Une méthode basée sur le théorème de Sturm
Internal Report, Centre de Recherche en Informatique de Nancy, Nancy, France, June 1988

[Se54] A. Seidenberg
A new decision method for elementary algebra
Annals of Mathematics 60 (2), September 1954, pp. 365-374

[Si89] Jörg Siekmann
Unification theory
Journal of Symbolic Computation 7, 1989, pp. 207-274

[SK90] Joachim Steinbach, Ulrich Kühler
Check your ordering - Examples and open problems
SEKI-Report, Artificial Intelligence Laboratories, Department of Computer Science, University of Kaiserslautern, West Germany, forthcoming

[St76] Mark E. Stickel
The inadequacy of primitive recursive complexity measures for determining finite termination of sets of reductions
Unpublished Memo, December 1976

[St89a] Joachim Steinbach
Extensions and comparison of simplification orderings
Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill, North Carolina, U.S.A., April 1989, LNCS 355, pp. 434-448

[St89b] Joachim Steinbach
Proving termination of associative-commutative rewriting systems using the Knuth-Bendix ordering
SEKI-Report SR-89-13, Artificial Intelligence Laboratories, Department of Computer Science, University of Kaiserslautern, West Germany, 1989

[St89c] Joachim Steinbach
COMTES - Vervollständigung von Termersetzungssystemen
SEKI-Working-Paper SWP-89-07, Artificial Intelligence Laboratories, Department of Computer Science, University of Kaiserslautern, West Germany, 1989

[St89d] Joachim Steinbach
Path and decomposition orderings for proving AC-termination
SEKI-Report SR-89-18, Artificial Intelligence Laboratories, Department of Computer Science, University of Kaiserslautern, West Germany, 1989

[St90a] Joachim Steinbach
Associative-commutative Knuth-Bendix ordering
Proceedings 2. Workshop "Termersetzung: Grundlagen und Anwendung", Dortmund, BRD, März 1990, Forschungsbericht Nr. 336, FB Informatik, Universität Dortmund, pp. 19-21

[St90b] Joachim Steinbach
Improving associative path orderings
Proceedings of the 10th Conference on Automated Deduction, Kaiserslautern, West Germany, July 1990, to appear

[Ta51] Alfred Tarski
A decision method for elementary algebra and geometry
University of California Press, Berkeley, 1951

[Th79] Paul R. Thie
An introduction to linear programming and game theory
John Wiley & Sons Inc., ISBN 0-471-04248-X, 1979

[Ze89] Michael Zehnter
Theorievollständige Ordnungen - Ein Überblick
Project Report, Department of Computer Science, University of Kaiserslautern, West Germany, November 1989

Appendix: Proofs

This supplement contains the proofs of the most important lemmata appearing in chapters 2, 3, 5, 6 and 7.

Proofs of chapter 2

Lemma 2.2.3

Let [...] be a polynomial interpretation for ground terms.
Then, the corresponding ordering $>_{POL}$ on Γ_G is compatible with the structure of terms if and only if each polynomial p ∈ [𝔉] is monotonous w.r.t. $[\Gamma_G]$.

Proof: (i) First of all, we will show that the compatibility of $>_{POL}$ guarantees that p ∈ [𝔉] is monotonous w.r.t. $[\Gamma_G]$. We have to prove that

[f](...,[s],...) > [f](...,[t],...) whenever [s] > [t]:

[s] > [t] ⇝ s $>_{POL}$ t
by definition of $>_{POL}$

⇝ f(...,s,...) $>_{POL}$ f(...,t,...)
by compatibility of $>_{POL}$

⇝ [f](...,[s],...) > [f](...,[t],...)
by definition of $>_{POL}$

(ii) Now, let each p ∈ [𝔉] be monotonous w.r.t. $[\Gamma_G]$.

s $>_{POL}$ t ⇝ [s] > [t]
by definition of $>_{POL}$

⇝ [f](...,[s],...) > [f](...,[t],...)
by precondition

⇝ f(...,s,...) $>_{POL}$ f(...,t,...)
by definition of $>_{POL}$ ∎

Theorem 2.2.9

Let [...] be a polynomial interpretation for ground terms.
Then, the corresponding ordering $>_{POL}$ on Γ_G is compatible with the structure of terms and well-founded if and only if each polynomial p ∈ [𝔉] is monotonous and non-diminishing on $[\Gamma_G]$.

Proof: According to lemma 2.2.3 the well-foundedness of $>_{POL}$ will be equivalent to all p ∈ [𝔉] being non-diminishing on $[\Gamma_G]$ if the monotony on $[\Gamma_G]$ of all p ∈ [𝔉] is presumed.

'$\rightsquigarrow$' : Suppose there exists an $f \in \mathfrak{F}$ such that $[t_i] > [f](\ldots,[t_i],\ldots)$ for any i. Since all p's are monotonous w.r.t. $[\Gamma_G]$, it follows that

$$[f](\ldots,[t_i],\ldots) > [f](\ldots,[f](\ldots,[t_i],\ldots),\ldots) > \ldots$$

Thus, $t_i >_{POL} f(\ldots,t_i,\ldots) >_{POL} \ldots$ is an infinitely descending chain in Γ_G. Therefore, $>_{POL}$ is not well-founded.

'$\leftsquigarrow$' : Suppose $>_{POL}$ is not well-founded. Then, $>_{POL}$ is self-embedding by the tree theorem of Kruskal. Similar to the proof of the embedding lemma in [De82], we will show by induction on $|s| + |t|$ that $s \sqsupset t$ implies $s \gtrsim_{POL} t$ for all ground terms $s = f(s_1,\ldots,s_m)$ and $t = g(t_1,\ldots,t_n)$, if all $p \in [\mathfrak{F}]$ are non-diminishing on $[\Gamma_G]$.

(i) $f = g \quad \wedge \quad s_i \sqsupset t_i$ for all $i \in [1,n]$

$\rightsquigarrow$ $s_i \gtrsim_{POL} t_i$ for $i \in [1,n]$
by induction hypothesis

$\rightsquigarrow$ $s = f(s_1,\ldots,s_n) \gtrsim_{POL} f(t_1,\ldots,t_n) = t$
using the monotony of $[f] \in [\mathfrak{F}]$ on $[\Gamma_G]$

(ii) $s_i \sqsupset t$ for at least one $i \in [1,n]$

$\rightsquigarrow$ $s_i \gtrsim_{POL} t$
by induction hypothesis

$\rightsquigarrow$ $[s_i] \geq [t]$
by definition of $>_{POL}$

$\rightsquigarrow$ $[f](\ldots,[s_i],\ldots) \geq [t]$
since $[f]$ is non-diminishing

$\rightsquigarrow$ $f(\ldots,s_i,\ldots) \gtrsim_{POL} t$
by definition of $>_{POL}$

Now we can show that $>_{POL}$ is not self-embedding. Suppose $>_{POL}$ to be self-embedding. Then, there exist t, t' such that $t >_{POL} t'$ and $t' \sqsupset t$. Since $\sqsupset \subseteq \gtrsim_{POL}$ and by definition of $>_{POL}$ we get

$$[t] > [t'] \quad \text{and} \quad [t'] \geq [t] \quad \lightning$$

Therefore, $>_{POL}$ is not self-embedding and noetherian. ∎

Lemma 2.2.10

Let $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$ be the set of all polynomials $p : \mathbb{R}^n \rightarrow \mathbb{R}$ of any arity $n \in \mathbb{N}_o$ that have coefficients of $[1, \infty]$. Let [...] be a polynomial interpretation for ground terms that attaches to each n-ary function symbol $f \in \mathfrak{F}$ a strict n-ary polynomial $p \in \mathfrak{Pol}(\mathbb{R}^{\geq 1})$.
Then, the corresponding ordering $>_{POL}$ is well-founded and compatible with the structure of terms.

Proof: The proof is straightforward using theorem 2.2.9. ∎

Lemma 2.2.11

Let [...] be a polynomial interpretation for ground terms that attaches

(1) a value $C \in \mathbb{R}$ to each constant function symbol $c \in \mathfrak{F}$ such that $C \geq 1$

(2) a strict n-ary polynomial p to each n-ary $(n > 0)$ function symbol $f \in \mathfrak{F}$ such that

$$p(X_1, \ldots, X_n) = \sum_{r_1, \ldots, r_n \in \mathbb{N}_o} c_{r_1 \ldots r_n} X_1^{r_1} \ldots X_n^{r_n}$$

such that for each i $(1 \leq i \leq n)$ $\sum_{\substack{r_i > o \\ r_1, \ldots, r_n \in \mathbb{N}_o}} c_{r_1 \ldots r_n} \geq 1$

Then, the corresponding ordering $>_{POL}$ on Γ_G is a reduction ordering.

Proof: The proof is straightforward using theorem 2.2.9. ∎

Theorem 2.3.3

Let [...] be a polynomial interpretation for variable terms that maps $\mathfrak{F}$ to $\mathfrak{Pol}(\mathbb{R})$.
The ordering defined by

$$s >_{POL} t \quad \text{iff} \quad [s] >_P [t]$$

is a reduction ordering on Γ that is stable by instantiations if and only if each polynomial $p \in [\mathfrak{F}]$ is monotonous and non-diminishing on $[\Gamma_G]$.

Proof: '⇝' : The interpreting polynomials are easily proved to be monotonous and non-diminishing on $[\Gamma_G]$ by contradiction.

'$\leftsquigarrow$' : $>_{POL}$ is irreflexive, transitive and stable by instantiations if $>_P$ has these properties. The monotony and the well-foundedness of $>_{POL}$ are valid since they hold on Γ_G and $>_{POL}$ is stable by instantiations. ∎

Lemma 2.3.4

Let [...] be a polynomial interpretation for variable terms that maps $\mathfrak{F}$ to $\mathfrak{Pol}(\mathbb{R}^{\geq 1})$.
The ordering defined by

$$s >_{POL} t \quad \text{iff} \quad [s] >_P [t]$$

is a reduction ordering on Γ that is stable by instantiations.

Proof: The proof is straightforward using theorem 2.3.3. ∎

Proofs of chapter 3

Suggestion 3.2.10

Let $l = f(s_1,...,s_m) \rightarrow g(t_1,...,t_n) = r$ be a rule such that $m > 1$, $n \geq 1$. There exists no separate interpretation [f] such that $l \geq_{POL} r$ if

$$(\exists s_i) \quad V(s_i) \cap \bigcup_{j=1, j\neq i}^{m} V(s_j) = \emptyset$$

$$\wedge \quad (\exists u \in O(r)) \;\; r/u = f(s_1',...,s_m') \quad \wedge \quad s_i' \sqsupset s_i$$

$$\wedge \quad V(r[u.i \leftarrow \lambda]) \cap V(s_i) \neq \emptyset$$

Proof: Assume that $[f](x_1,...,x_m) = \sum_{i=1}^{m} p_i(x_i)$ where $p_i(x_i)$ is the polynomial consisting of x_i, only.

$\leadsto$ $[f(s_1,...,s_m)] = \sum_{j=1}^{m} p_j(s_j)$

Note that $s_i' \sqsupset s_i$ which implies that $p_i(s_i') \geq_{POL} p_i(s_i)$.

$\leadsto$ The grade of the variables of $V(s_i)$ belonging to [r] is greater than or equal to that belonging to [l] because the grade of the variables of $V(s_i)$ belonging to $[f(t_1,...,t_m)]$ is equal to that of $p_i(t_i)$ since [f] is separate.

Note that there is any variable x of $V(s_i)$ occurring in r outside of s_i' (by precondition).

$\leadsto$ The grade of x in [r] is greater than that of x in [l]

$\leadsto$ $\neg(l \geq_{POL} r)$ ■

Lemma 3.3.2

Let $a = \sum_{i=1}^{n} a_i x_i + a_0$ and $b = \sum_{i=1}^{n} b_i x_i + b_0$ be linear polynomials with $(\forall i \in [1,n])\; a_i, b_i \geq 0$. Then,

$$a \geq_P b$$
$$\text{if} \quad (\forall i \in [0,n]) \;\; a_i \geq b_i$$

Proof: $a_0 + a_1x_1 + ... + a_nx_n \geq_P b_0 + b_1x_1 + ... + b_nx_n$
iff
$(a_0 - b_0) + (a_1 - b_1)x_1 + ... + (a_n - b_n)x_n \geq_P 0$

This inequality is valid if $(\forall i \in [0,n])\; a_i \geq b_i$ ■

Lemma 3.3.4

$$\prod_{i=1}^{n} a_i \geq \sum_{i=1}^{n} a_i$$
if $(\forall i \in [1,n])\ a_i \geq 2$

Proof: We will prove this fact by means of contradiction. Assume that

$$a_1 \cdot \ldots \cdot a_n < a_1 + \ldots + a_n$$

$$\rightsquigarrow \quad 1 < \frac{a_1}{a_1 \cdot \ldots \cdot a_n} + \ldots + \frac{a_n}{a_1 \cdot \ldots \cdot a_n}$$

$$\rightsquigarrow \quad 1 < \frac{1}{a_2 \cdot \ldots \cdot a_n} + \ldots + \frac{1}{a_1 \cdot \ldots \cdot a_{n-1}}$$

Note that the right-hand side of this inequality is not greater than

$$\frac{1}{(n-1) \cdot 2} \cdot n$$

since $a_i \geq 2$

$$\rightsquigarrow \quad 1 < \frac{n}{(n-1) \cdot 2}$$

$$\rightsquigarrow \quad ↯$$

because $\frac{n}{(n-1) \cdot 2} \leq 1$ for $n > 1$

(The case $n = 1$ is obvious since $a_1 \geq a_1$) ■

Lemma 3.3.5

$$n \cdot \prod_{i=1}^{n} a_i \geq \sum_{i=1}^{n} a_i$$
if $(\forall i \in [1,n])\ a_i \geq 1$

Proof: We will prove this lemma by induction on n.

- The base case $(n = 1)$ is obvious.

- $n \rightarrow n+1$:

$$(n+1)a_1 \cdot \ldots \cdot a_n \cdot a_{n+1} = n \cdot a_1 \cdot \ldots \cdot a_n \cdot a_{n+1} + a_1 \cdot \ldots \cdot a_n \cdot a_{n+1}$$

$$\geq (a_1 + \ldots + a_n) \cdot a_{n+1} + a_1 \cdot \ldots \cdot a_n \cdot a_{n+1}$$

by using the induction hypothesis

$$= a_1 \cdot a_{n+1} + \ldots + a_n \cdot a_{n+1} + a_1 \cdot \ldots \cdot a_n \cdot a_{n+1}$$

$$\geq a_1 + \ldots + a_n + a_{n+1}$$

since $a_i \cdot a_{n+1} \geq a_i \quad \wedge \quad a_1 \cdot \ldots \cdot a_n \cdot a_{n+1} \geq a_{n+1}$
(because $a_i \geq 1$) ■

Lemma 3.3.6

$$\frac{1}{n} \cdot \sum_{i=1}^{n} a_i \geq \left(\prod_{i=1}^{n} a_i \right)^{\frac{1}{n}}$$

if $(\forall i \in [1,n]) \; a_i \geq 0$

Proof: The proof of this lemma can be found in [BB61] on page 54. ■

Lemma 3.3.7

Let $a_i \geq 1$ and $b_i \geq 1$. Then,

$$\sum_{i=1}^{n} a_i \geq \sum_{i=1}^{m} b_i$$

if i) $n^n \cdot \prod_{i=1}^{n} a_i \geq \prod_{i=1}^{m} b_i^n \quad \wedge \quad (\forall i \in [1,m]) \; b_i \geq 2$

or ii) $n^n \cdot \prod_{i=1}^{n} a_i \geq m^n \cdot \prod_{i=1}^{m} b_i^n$

Proof: This lemma describes two estimations of the inequality $\sum_{i=1} a_i \geq \sum_{i=1} b_i$, i.e. we found a and b such that $\sum_{i=1}^{n} a_i \geq a \overset{!}{\geq} b \geq \sum_{i=1}^{m} b_i$. It is obvious that $a \geq b$ implies $\sum_{i=1}^{n} a_i \geq \sum_{i=1}^{m} b_i$.

i) $\sum_{i=1}^{n} a_i \overset{(*)}{\geq} n \cdot \left(\prod_{i=1}^{n} a_i \right)^{\frac{1}{n}} \overset{!}{\geq} \prod_{i=1}^{m} b_i \overset{(**)}{\geq} \sum_{i=1}^{m} b_i$

with (*) = lemma 3.3.6 and (**) = lemma 3.3.4

$\rightsquigarrow \quad n^n \cdot \prod_{i=1}^{n} a_i \overset{!}{\geq} \prod_{i=1}^{m} b_i^n$

Note that it must be guaranteed that $(\forall i \in [1,m]) \; b_i \geq 2$ because it is a precondition for the use of (**).

ii) $\sum_{i=1}^{n} a_i \overset{(*)}{\geq} n \cdot (\prod_{i=1}^{n} a_i)^{\frac{1}{n}} \overset{!}{\geq} m \cdot \prod_{i=1}^{m} b_i \overset{(**)}{\geq} \sum_{i=1}^{m} b_i$

with (*) = lemma 3.3.6 and (**) = lemma 3.3.5

$$\rightsquigarrow \quad n^n \cdot \prod_{i=1}^{n} a_i \overset{!}{\geq} m^n \cdot \prod_{i=1}^{m} b_i^n$$ ∎

Lemma 3.3.11

$$\prod_{i=1}^{n} a_i \geq \prod_{i=1}^{m} b_i$$

$$\text{iff} \quad \sum_{i=1}^{n} lb(a_i) \geq \sum_{i=1}^{m} lb(b_i)$$

Proof: $\prod_{i=1}^{n} a_i \geq \prod_{i=1}^{m} b_i$

iff

$a_1 \cdot \ldots \cdot a_n \geq b_1 \cdot \ldots \cdot b_m$

iff

$lb(a_1 \cdot \ldots \cdot a_n) \geq lb(b_1 \cdot \ldots \cdot b_m)$

iff

$lb(a_1) + \ldots + lb(a_n) \geq lb(b_1) + \ldots + lb(b_m)$

iff

$\sum_{i=1}^{n} lb(a_i) \geq \sum_{i=1}^{m} lb(b_i)$ ∎

Theorem 3.3.15

The algorithm 3.3.13 always terminates. If it does not fail $\mathfrak{R}$ can be ordered by a polynomial ordering (using only linear polynomials).

Proof: It is easy to see that the transformation process of each rule ((1) - (8)) terminates. Therefore, the algorithm 3.3.13 terminates.
The correctness of the algorithm is guaranteed by lemmata 3.3.2, 3.3.4 - 3.3.7 and 3.3.11. ∎

Proofs of chapter 5

Lemma 5.2.1

$>_{POL}$ is E-compatible if
$(\forall s, t \in \Gamma)\ \ s =_E t \ \ \rightsquigarrow \ \ s =_{POL} t$

Proof: We have to prove

$$s' =_E s >_{POL} t =_E t' \quad \rightsquigarrow \quad s' >_{POL} t'.$$

Note that

$$s' =_E s \ \wedge \ t =_E t' \quad \rightsquigarrow \quad s' =_{POL} s \ \wedge \ t =_{POL} t'$$

which implies

$$s' =_{POL} s >_{POL} t =_{POL} t'.$$

Thus, the assertion holds since $t =_{POL} t'$ implies $[t] = [t']$. ■

Lemma 5.4.1

$C[f]$
iff
$[f](x, y) = \sum_{i,j} a_{ij} x^i y^j$
$\wedge \ \ (\forall i, j)\ a_{ij} = a_{ji}$

Proof: Let be $[f](x, y) = \sum a_{ij} x^i y^j$. Then,

$$[f(x, y)] = \sum_{i \geq 0} a_{ii} x^i y^i + \sum_{i > j \geq 0} a_{ij} x^i y^j + \sum_{j > i \geq 0} a_{ij} x^i y^j$$

$$[f(y, x)] = \sum_{i \geq 0} a_{ii} x^i y^i + \sum_{i > j \geq 0} a_{ij} x^j y^i + \sum_{j > i \geq 0} a_{ij} x^j y^i$$

We have to distinguish between two cases:

i) $a_{ii} > 0$

$$\rightsquigarrow \quad a_{ii} x^i y^i \in [f(x, y)] \ \ \wedge \ \ a_{ii} y^i x^i \in [f(y, x)]$$

ii) $a_{ij} > 0$ and $i \neq j$

$$\rightsquigarrow \quad a_{ij} x^i y^j \in [f(x, y)] \ \ \wedge \ \ a_{ij} y^i x^j \in [f(y, x)]$$

Since $i \neq j$, $a_{ij}x^i y^j \neq a_{ij}x^j y^i$. Therefore, we have to require a monomial $a_{ji}x^j y^i$ for each $a_{ij}x^i y^j$. ∎

Lemma 5.4.2

There is no interpretation of f such that
$C(f) \wedge [f(f(x,y),z) >_{POL} f(x,f(y,z)) \vee f(x,f(y,z)) >_{POL} f(f(x,y),z)]$

Proof: Obvious, since a commutative interpretation (which is necessary for C(f)) implies the terms f(f(x, y), z) and f(x, f(y, z)) to be equivalent or incomparable. ∎

Lemma 5.4.4

$$C_L(f) \quad \text{if} \quad [f](x,y) = \sum_i a_i x^j y^i + (1-j)x \;\wedge\; j \in \{0,1\}$$
$$\text{or} \quad [f](x,y) = (x+1)\sum_{i\geq 1} a_i y^i + a_0 x + a_0 - 1$$

Proof: i) Let $[f](x,y) = a_0 + a_1 y + \ldots + a_1 y^n + x$.

$$\leadsto \quad [f(f(x,y),z)] = a_0 + a_1 z + \ldots + a_n z^n + a_0 + a_1 y + \ldots + a_1 y^n + x$$
$$[f(f(x,z),y)] = a_0 + a_1 y + \ldots + a_n y^n + a_0 + a_1 z + \ldots + a_n z^n + x$$

$$\leadsto \quad [f(f(x,y),z)] = [f(f(x,z),y)]$$

ii) Let $[f](x,y) = x(a_0 + a_1 y + \ldots + a_n y^n)$

$$\leadsto \quad [f(f(x,y),z)] = x(a_0 + a_1 y + \ldots + a_n y^n)(a_0 + a_1 z + \ldots + a_n z^n)$$
$$[f(f(x,z),y)] = x(a_0 + a_1 z + \ldots + a_n z^n)(a_0 + a_1 y + \ldots + a_n y^n)$$

$$\leadsto \quad [f(f(x,y),z)] = [f(f(x,z),y)]$$

iii) Let $[f](x,y) = (x+1)\sum_{i\geq 1} a_i y^i + a_0 x + a_0 - 1$

$$\leadsto \quad [f(f(x,y),z)] = ((x+1)\sum_{i\geq 1} a_i y^i + a_0 x + a_0 - 1 + 1)\sum_{i\geq 1} a_i z^i + a_0(x+1)\sum_{i\geq 1} a_i y^i + a_0^2 x + a_0^2 - a_0 + a_0 - 1$$
$$= (x+1)\sum_{i\geq 1} a_i y^i \sum_{i\geq 1} a_i z^i + a_0 x \sum_{i\geq 1} a_i z^i + a_0 \sum_{i\geq 1} a_i z^i + a_0(x+1)\sum_{i\geq 1} a_i y^i + a_0^2 x + a_0^2 - 1$$

$$[f(f(x,z),y)] = ((x+1)\sum_{i\geq 1} a_i z^i + a_0 x + a_0 - 1 + 1)\sum_{i\geq 1} a_i y^i + a_0(x+1)\sum_{i\geq 1} a_i z^i + a_0^2 x + a_0^2 - a_0 + a_0 - 1$$

$$= (x + 1)\sum_{i\geq 1} a_i z^i \sum_{i\geq 1} a_i y^i + a_0 x \sum_{i\geq 1} a_i y^i + a_0 \sum_{i\geq 1} a_i y^i + a_0(x + 1) \sum_{i\geq 1} a_i z^i + a_0^2 x + a_0^2 - 1$$

$\leadsto$ $[f(f(x, y), z)] = [f(f(x, z), y)]$ ∎

Lemma 5.4.6

$C_R(f)$ if $[f](x, y) = \sum_i a_i x^i y^j + (1 - j)y$
$\wedge$ $j \in \{0, 1\}$
or $[f](x, y) = (y + 1) \sum_{i\geq 1} a_i x^i + a_0 y + a_0 - 1$

Proof: Analogous with the proof of lemma 5.4.4 ∎

Lemma 5.4.8

- $C(f) \wedge C_L(f)$ if $[f](x, y) = axy + b(x + y) + c$ and $b = c = 0 \vee [a = 0 \wedge b = 1] \vee [a = b \wedge c = b - 1]$
- $C(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$
- $C_L(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$
- $C(f) \wedge C_L(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$

Proof: i) - Let $[f](x, y) = axy$

$\leadsto$ $[f(x, y)] = axy = ayx = [f(y, x)]$
$[f(f(x, y), z)] = a(axy)z = a(axz)y = [f(f(x, z), y)]$

- Let $[f](x, y) = x + y + c$

$\leadsto$ $[f(x, y)] = x + y + c = y + x + c = [f(y, x)]$
$[f(f(x, y), z)] = x + y + c + z + c = x + z + c + y + c = [f(f(x, z), y)]$

- Let $[f](x, y) = bxy + b(x + y) + b - 1$

$\leadsto$ $[f(x, y)] = bxy + b(x + y) + b - 1 = byx + b(y + x) + b - 1 = [f(y, x)]$
$[f(f(x, y), z)] = b(bxy + b(x + y) + b - 1)z + b(bxy + b(x + y) + b - 1 + z) + b - 1 = b^2xyz + b^2xz$

$$+ b^2yz + b^2z - bz + b^2xy + b^2x + b^2y$$
$$+ b^2 - b + bz + b - 1$$
$$= b(bxz + b(x + z) + b - 1)y + b(bxz + b(x + z)$$
$$+ b - 1 + y) + b - 1 = [f(f(x,z),y)]$$

ii) $C(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$:
analogous with i)

iii) $C_L(f) \wedge C_R(f)$ if $C(f) \wedge C_L(f)$:
analogous with i)

iv) $C(f) \wedge C_L(f) \wedge C_R(f)$
if $C(f) \wedge C(f) \wedge C_L(f)$
(with the help of iii))

iff $C(f) \wedge C_L(f)$ ■

Lemma 5.5.1

$D_L(f,g)$ if $[f](x,y) = \sum_i a_i xy^i$, $[g](x,y) = b_1x + b_2y$
or $[f](x,y) = a_1xy + a_2(x + y) + a_3$, $[g](x,y) = x + y + b_3$
$\wedge$ $a_3 = b_3(a_1b_3 - 1)$, $a_2 = a_1b_3$

Proof: i) Let $[f](x,y) = \sum_i a_i xy^i$, $[g](x,y) = b_1x + b_2y$

$\rightsquigarrow$ $[f(g(x,y),z)] = \sum_i a_i(b_1x + b_2y)z^i = \sum_i b_1a_ixz^i + \sum_i b_2a_iyz^i$
$= b_1\sum_i a_ixz^i + b_2\sum_i a_iyz^i = [g(f(x,z),f(y,z))]$

ii) Let $[f](x,y) = a_1xy + a_1b_3(x + y) + b_3(a_1b_3 - 1)$, $[g](x,y) = x+y+b_3$

$\rightsquigarrow$ $[f(g(x,y),z)] = a_1(x + y + b_3)z + a_1b_3(x + y + b_3 + z) +$
$+ b_3(a_1b_3 - 1) = a_1xz + a_1yz + a_1b_3z + a_1b_3x + a_1b_3y + a_1b_3^2 + a_1b_3z$
$+ b_3(a_1b_3 - 1) = a_1xz + a_1b_3(x + z) + b_3(a_1b_3 - 1) + a_1yz + a_1b_3(y+z) +$
$+ b_3(a_1b_3 - 1) + b_3 = [g(f(x,z),f(y,z))]$ ■

Lemma 5.5.3

$D_R(f,g)$ if $[f](x,y) = \sum_i a_i x^i y$, $[g](x,y) = b_1x + b_2y$
or $[f](x,y) = a_1xy + a_2(x + y) + a_3$, $[g](x,y) = x + y + b_3$
$\wedge$ $a_3 = b_3(a_1b_3 - 1)$, $a_2 = a_1b_3$

Proof: Analogous with the proof of lemma 5.5.1 ■

Lemma 5.5.5

- $D_L(f,g) \wedge D_R(f,g)$ if $[f](x,y) = axy$, $[g](x,y) = bx + cy$
  or $[f](x,y) = axy + ad(x + y) + d(ad - 1)$,
  $[g](x,y) = x + y + d$

- The following combinations of theories are not allowed, i.e. there exists no polynomial ordering which induces the equality of both sides of the axioms:

  $D_L(f,g) \wedge D_R(g,f)$
  $D_L(f,g) \wedge D_L(g,f)$
  $D_R(f,g) \wedge D_R(g,f)$

Proof: i) - Let $[f](x,y) = axy$, $[g](x,y) = bx + cy$

$$\leadsto \quad [f(g(x,y),z)] = a(bx + cy)z = $$
$$baxz + cayz = [g(f(x,z),f(y,z))]$$
$$[f(x,g(y,z))] = ax(by + cz) = $$
$$baxy + caxz = [g(f(x,y),f(x,z))]$$

- Let $[f](x,y) = axy + ad(x + y) + d(ad - 1)$, $[g](x,y) = x + y + d$

$$\leadsto \quad [f(g(x,y),z)] = a(x + y + d)z + ad(x + y + d + z) + d(ad - 1)$$
$$= axz + ayz + adz + adx + ady + ad^2 + adz + ad^2 - d$$
$$= axz + ad(x + z) + d(ad - 1) + ayz + ad(y + z) + d(ad - 1) + d$$
$$= [g(f(x,z),f(y,z))]$$

ii) $D_L(f,g) \wedge D_R(g,f)$:

The equalities $[f(g(x,y),z)] = [g(f(x,z),f(y,z))]$
and $[g(x,f(y,z))] = [f(g(x,y),g(x,z))]$
imply the equality

$$[f(g(x,y),z)] = [f(g(f(x,z),y),g(f(x,z),z))]$$

(gained by unifying $g(f(x,z),f(y,z))$ and $g(x,f(y,z))$).
Thus, $[f(g(x,x),x)] \overset{!}{=} [f(g(f(x,x),x),g(f(x,x),x))]$.
By a relatively simple analysis one can prove that there are no polynomials interpreting f and g such that the above equality is valid.

iii) $D_L(f,g) \wedge D_L(g,f)$:

The considerations of i) will lead to the equality

$$[f(g(x,x),x)] = [f(g(x,f(x,x)),g(x,f(x,x)))]$$

which cannot be verified with the help of any polynomial ordering.

iv) $D_R(f, g) \wedge D_R(g, f)$:
analogous with iii) ∎

Lemma 5.6.1

$$E(f, g) \quad \text{if} \quad [f](x) = x \ , \ [g](x, y) = \sum_{j,k} a_{jk} x^j y^k$$
$$\text{or} \ [f](x) = a_1 x + a_2 \ , \ [g](x, y) = b_1 x + b_2 y + b_3 \ \wedge \ b_3(a_1 - 1) = a_2(b_1 + b_2 - 1)$$
$$\text{or} \ [f](x) = a_1 x^i \ , \ [g](x, y) = b_1 x^j y^k \ \wedge \ a_1^{j+k-1} = b_1^{i-1}$$

Proof: i) Let $[f](x) = x$ and $[g](x, y) = \sum_{j,k} a_{jk} x^j y^k$

$$\rightsquigarrow \ [f(g(x, y))] = \sum_{j,k} a_{jk} x^j y^k = [g(f(x), f(y))]$$

ii) Let $[f](x) = a_1 x + a_2$ and $[g](x, y) = b_1 x + b_2 y + b_3$ such that $b_3(a_1 - 1) = a_2(b_1 + b_2 - 1)$

$$\rightsquigarrow \ \begin{aligned}[f(g(x, y))] &= a_1 b_1 x + a_1 b_2 y + a_1 b_3 + a_2 \\ [g(f(x), f(y))] &= a_1 b_1 x + a_1 b_2 y + a_2 b_1 + a_2 b_2 + b_3\end{aligned}$$

$$\rightsquigarrow \ a_2 + a_1 b_3 \overset{!}{=} a_2 b_1 + a_2 b_2 + b_3$$

$$\leftrightsquigarrow \ a_1 b_3 - b_3 \overset{!}{=} a_2 b_1 + a_2 b_2 - a_2$$

which is equivalent to the precondition

iii) Let $[f](x) = a_1 x^i$ and $[g](x, y) = b_1 x^j y^k$ such that $a_1^{j+k-1} = b_1^{i-1}$.

$$\rightsquigarrow \ \begin{aligned}[f(g(x, y))] &= a_1 b_1^i x^{ij} y^{ik} \\ [g(f(x), f(y))] &= a_1^{j+k} b_1 x^{ij} y^{ik}\end{aligned}$$

$$\rightsquigarrow \ a_1 b_1^i \overset{!}{=} a_1^{j+k} b_1$$

$$\leftrightsquigarrow \ b_1^{i-1} \overset{!}{=} a_1^{j+k-1}$$

which represents the precondition ∎

Lemma 5.6.2

$[f(g(x,y))] = [g(f(y),f(x))]$

if $[f](x) = x \quad , \quad [g](x,y) = \sum_{j,k} a_{jk} x^j y^k$

$\wedge \quad (\forall j,k)\ a_{jk} = a_{kj}$

or $[f](x) = a_1 x + a_2 \quad , \quad [g](x,y) = b_1(x + y) + b_2$

$\wedge \quad b_2(a_1 - 1) = a_2(2b_1 - 1)$

or $[f](x) = a_1 x^i \quad , \quad [g](x,y) = b_1 x^j y^j$

$\wedge \quad a_1^{2j-1} = b_1^{i-1}$

Proof: Analogous with the proof of lemma 5.6.1 ∎

Lemma 5.8.1

$M(f)$

iff

$[f](x) = x$

Proof: Let $[f](x) = ax^i + \ldots$ such that i is the greatest exponent. Then,

$[f(f(x))] = a^{i+1}x^{i^2} + \ldots$

$[x] \quad = x$

$\rightsquigarrow \quad a^{i+1}x^{i^2} \overset{!}{=} x$

$\rightsquigarrow \quad i = 1 \quad \wedge \quad a = 1$

$\rightsquigarrow \quad [f](x) = x + c$

$\rightsquigarrow \quad x + 2c \overset{!}{=} x$

$\rightsquigarrow \quad c = 0$

$\rightsquigarrow \quad [f](x) = x$ ∎

Lemma 5.8.3

$M(f) \quad \wedge \quad [f(g(x,y))] = [g(f(y),f(x))]$

if $[f](x) = x \quad , \quad [g](x,y) = \sum_{i,j} a_{ij} x^i y^j$ and $(\forall i,j)\ a_{ij} = a_{ji}$

Proof: $[f](x) = x$ (by using lemma 5.8.1)

$\leadsto$ $[f(g(x,y))] = [g(x,y)] \quad \wedge \quad [g(f(y),f(x))] = [g(y,x)]$

$\leadsto$ $[g]$ must be commutative (see lemma 5.4.1) ■

Lemma 5.11.1

$U_L(f,1)$
iff
$[f](x,y) = x^i y \quad , \quad [1](\,) = 1$

Proof: Let $[f](x,y) = \sum_{i,j\geq 0} a_{ij}x^i y^j$ and $[1](\,) = b$. Then,

$[f(1,x)] = \sum_{i,j\geq 0} a_{ij}b^i x^j \overset{!}{=} x = [x]$

$\leadsto$ $(\forall j \neq 1) \quad a_{ij}b^i = 0$

$\leadsto$ $[f](x,y) = \sum_{i\geq 1} a_i x^i y$

$\leadsto$ $\sum_{i\geq 1} a_i b^i = 1$

Let $a_i, b, i \in \mathbb{N}$

$\leadsto$ $(\exists i)\ a_i b^i = 1 \quad \wedge \quad (\forall j \neq i)\ a_j b^j = 0$

$\leadsto$ $a_i = 1 \quad \wedge \quad b^i = 1 \quad \wedge \quad a_j = 0$
since $b^j \neq 0$

$\leadsto$ $a_i = 1 \quad \wedge \quad b = 1$
since $i > 0$

$\leadsto$ $[f](x,y) = x^i y$
$[1](\,) = 1$ ■

Lemma 5.11.2

$U_R(f,1)$
iff
$[f](x,y) = xy^i \quad , \quad [1](\,) = 1$

Proof: Let $[f](x,y) = \sum_{i,j\geq 0} a_{ij}x^i y^j$ and $[1](\,) = b$. Then,

$[f(x,1)] = \sum_{i,j\geq 0} a_{ij}b^j x^i \overset{!}{=} x = [x]$

$\leadsto$ $(\forall i \neq 1)\ a_{ij}b^j = 0$

$\leadsto$ $[f](x,y) = \sum_{j\geq 1} a_j x y^j$

$\leadsto$ $\sum_{j\geq 1} a_j b^j = 1$

$\leadsto$ $[f](x,y) = xy^j$
$[1]() = 1$

analogous with the proof of lemma 5.11.1. ∎

Lemma 5.11.4

$U_L(f,1) \ \wedge\ U_R(f,1)$
if $[f](x,y) = xy$, $[1]() = 1$

Proof: Let $[f](x,y) = x^i y = xy^i$ (due to lemma 5.11.1 and lemma 5.11.2)

$\leadsto$ $i = 1$ ∎

For compactness, the proofs of lemmata 5.12.1 - 5.12.8 about the combinations of underlying theories are not given here.

Lemma 5.12.11

The algorithm 5.12.9 always terminates. It provides restrictions for interpretations that have to be met by combinations of up to three different theories.

Proof: The proof of the termination is obvious since there are no cycles in the lemmata of chapter 5.
The correctness of the algorithm is guaranteed by the correctness of the lemmata of chapter 5. ∎

Lemma 5.13.2

There are no interpretations of function symbols such that both sides of a variable-reducing equation are equivalent w.r.t. a polynomial ordering.

Proof: Let $V(s) = V(t) \,\dot\cup\, \{x\}$, $s/u = f(\ldots)$ and $s/ui = x$. In order to obtain $[s] = [t]$ we have to set the coefficient of the i-th argument of $[f]$ to zero. This is a contradiction to the monotony of the polynomials. ∎

Lemma 5.13.4

There are no interpretations of function symbols such that both sides of a subterm-minimizing equation are equivalent w.r.t. a polynomial ordering.

Proof: W.l.o.g. let $s = f(\ldots, t, \ldots)$.

Assume that there are interpretations such that $[s] = [t]$.

$\leadsto$ We have to set the coefficient of at least one argument of $[f]$ to zero since the operator f has more than one argument (because the equation is subterm-minimizing).

$\leadsto$ contradiction to the monotony of the polynomials ∎

Lemma 5.13.5

There are no interpretations of the function symbol *if* such that both sides of each of the following equations are equivalent w.r.t. polynomial orderings:

$$
\begin{aligned}
if(x, y, y) &= y \\
if(x, x, false) &= x \\
if(x, true, x) &= x \\
if(x, if(x, y, z), z) &= if(x, y, z) \\
if(x, y, if(x, y, z)) &= if(x, y, z)
\end{aligned}
$$

Proof: The first equation is variable-reducing and the other ones are subterm-minimizing. Due to lemmata 5.13.2 and 5.13.4 the assertion holds. ∎

Lemma 5.13.6

$[if(if(x, y, z), u, v)] = [if(x, if(y, u, v), if(z, u, v))]$

if

$[if](x, y, z) = a_1xy + a_2xz + a_3x + a_4y + a_5z + a_6$

$\wedge$ • $a_1, a_2 > 0 \;\wedge\; a_3 = a_4 = a_5 = a_6 = 0$

or • $a_3, a_4, a_5, a_6 > 0 \;\wedge$

$a_3 = a_4 + a_5$

$a_1a_5 = a_2a_4$

$a_1a_6 = a_4(a_4 + a_5 - 1)$

Proof: i) Let $[if](x, y, z) = a_1xy + a_2xz$.

$\leadsto$ $[if(if(x, y, z), u, v)] = a_1(a_1xy + a_2xz)u + a_2(a_1xy + a_2xz)v$

$$= a_1^2xyu + a_1a_2xzu + a_1a_2xyv + a_2^2xzv = a_1x(a_1yu + a_2yv) + a_2x(a_1zu + a_2zv) = [if(x, if(y, u, v), if(z, u, v)]]$$

ii) Let $[if(x, y, z] = (a_1y + \frac{a_1a_5}{a_4}z + a_4 + a_5)x + a_4y + a_5z + \frac{a_4}{a_1}(a_5 + a_4 - 1).$

$$\rightsquigarrow [if(if(x, y, z), u, v)] = (a_1u + \frac{a_1a_5}{a_4}v + a_4 + a_5)\cdot((a_1y + \frac{a_1a_5}{a_4}z + a_4 + a_5)x + a_4y + a_5z + \frac{a_4}{a_1}(a_5 + a_4 - 1)) + a_4u + a_5v + \frac{a_4}{a_1}(a_5 + a_4 - 1)$$

$$= a_1^2xyu + a_1^2\frac{a_5}{a_4}xyv + a_1^2\frac{a_5}{a_4}xzu + a_1^2\frac{a_5^2}{a_4^2}xzv + a_1a_4xy + a_1a_5xy + a_1a_5xz + a_1\frac{a_5^2}{a_4}xz + a_1a_4xu + a_1a_5xu + a_1a_5xv + a_1\frac{a_5^2}{a_4}xv + a_1a_4yu + a_1a_5yv + a_1a_5zu + a_1\frac{a_5^2}{a_4}zv + a_4^2x + 2a_4a_5x + a_5^2x + a_4^2y + a_4a_5y + a_4a_5y + a_5^2z + a_4^2u + a_4a_5u + a_4a_5v + a_5^2v + 2a_4^2\frac{a_5}{a_1} + \frac{a_4^3}{a_1} + a_4\frac{a_5^2}{a_1} - \frac{a_4}{a_1}$$

$$= a_1x((a_1u + \frac{a_1a_5}{a_4}v + a_4 + a_5)y + a_4u + a_5v + \frac{a_4}{a_1}(a_5 + a_4 - 1)) + \frac{a_1a_5}{a_4}x((a_1u + \frac{a_1a_5}{a_4}v + a_4 + a_5)z + a_4u + a_5v + \frac{a_4}{a_1}(a_5 + a_4 - 1)) + a_4x + a_5x + a_4((a_1u + \frac{a_1a_5}{a_4}v + a_4 + a_5)y + a_1u + a_5v + \frac{a_4}{a_1}(a_5 + a_4 - 1)) + a_5((a_1u + \frac{a_1a_5}{a_4}v + a_4 + a_5)z + a_4u + a_5v + \frac{a_4}{a_1}(a_5 + a_4 - 1)) + \frac{a_4}{a_1}(a_5 + a_4 - 1) = [if(x, if(y, u, v), if(z, u, v)]] \blacksquare$$

Lemma 5.13.7

$[if(x, if(y, z, u), if(y, v, w)]] = [if(y, if(x, z, v), if(x, u, w)]]$
if
$[if](x, y, z) = \sum_i (a_ix^iy + b_ix^iz) + c_1y$

Proof: Let $[if](x, y, z) = \sum_i (a_ix^iy + b_ix^iz) + c_1y.$

$$\rightsquigarrow [if(x, if(y, z, u), if(y, v, w)]] = z\sum_i(a_ix^i\sum_i(a_iy^i)) + u\sum_i(a_ix^i\sum_i(b_iy^i)) + c_1z\sum_i(a_ix^i) + v\sum_i(b_ix^i\sum_i(a_iy^i)) + w\sum_i(b_ix^i\sum_i(b_iy^i)) + c_1v\sum_i(b_ix^i) + c_1z\sum_i(a_iy^i) + c_1u\sum_i(b_iy^i) + c_1^2z = [if(y, if(x, z, v), if(x, u, w)]].$$

$\blacksquare$

Lemma 5.13.9

Both sides of the axioms belonging to $\mathfrak{R}_{AG} \backslash \{x + i(x) = 0\}$ are equivalent w.r.t. the polynomial ordering based on the interpretations

$$[+](x, y) = xy$$
$$[i](x) = x$$
$$[0]() = 1$$

Proof: Obvious ∎

Lemma 5.13.10

There are no interpretations of function symbols such that both sides of the axioms of a quasi-group (see p. 102) are equivalent w.r.t. a polynomial ordering.

Proof: All axioms of a quasi-group are variable-reducing.

$\leadsto$ assertion since lemma 5.13.2 is valid ∎

Lemma 5.13.13

It will be impossible to use any part of the boolean ring $\mathfrak{R}_{BR}$ as an underlying theory (and the remaining part as reduction rules), if the termination is to be proved by a polynomial ordering.

Proof: Note that $\mathfrak{R}_{BR}$ includes

$$D_L(+, *) \text{ , } D_R(+, *) \text{ , } D_L(*, +) \text{ and } D_R(*, +)$$

as axioms or rules.

Due to lemma 5.5.5 the following combinations are not allowed as underlying theories:

$$D_L(+, *) \wedge D_R(*, +)$$
$$D_L(*, +) \wedge D_R(+, *)$$
$$D_L(+, *) \wedge D_L(*, +)$$
$$D_R(+, *) \wedge D_R(*, +)$$

Therefore, either $D_L(+, *) \wedge D_R(+, *)$ or $D_L(*, +) \wedge D_R(*, +)$ must be part of the reduction rule set. However, both requirements are impossible:

i) $(x * y) + z \;>_{POL}\; (x + z) * (y + z)$

$x + (y * z) \;>_{POL}\; (x + y) * (x + z)$

$[(x + y) * z] = [(x * z) + (y * z)]$

$[x * (y + z)] = [(x * y) + (x * z)]$

In order to orient the first two rules we have to require that $[*](x, y)$ must be a mixed interpretation and $[+](x, y)$ must be a separate one (due to suggestion 3.2.10). However, the orientation of the last two rules requires a mixed polynomial for $[+](x, y)$ and a separate one for $[*](x, y)$ (due to suggestion 3.2.10). This is a contradiction.

ii) $[(x * y) + z] = [(x + z) * (y + z)]$

$[x + (y * z)] = [(x + y) * (x + z)]$

$(x + y) * z \;>_{POL}\; (x * z) + (y * z)$

$x * (y + z) \;>_{POL}\; (x * y) + (x * z)$

analogous with i) ■

Proofs of chapter 6

Theorem 6.2.2

The improved polynomial ordering $>_{IPOL}$ is a simplification ordering on Γ and it is stable by instantiations.

Proof: For being a simplification ordering we have to prove the irreflexivity, the transitivity, the subterm property and the monotony.

A) $>_{IPOL}$ is irreflexive:
We have to show that $(\forall t \in \Gamma)\ t \not>_{IPOL} t$ which is true since $>_{POL}$ is irreflexive.

B) $>_{IPOL}$ is transitive:
We must prove that $r >_{IPOL} s >_{IPOL} t \leadsto r >_{IPOL} t$.
Proving this assertion by induction we have to consider three disjoint cases.

i) $[r] >_P [s] \quad \vee \quad [s] >_P [t]$

$\leadsto$ $[r] >_P [t]$
since $>_P$ is transitive and the fact that $[s] = [r] >_P [t] \vee [s] >_P [r] = [t]$ implies $[s] >_P [t]$

ii) $[r] = [s] = [t] \quad \wedge \quad (top(r) \triangleright top(s) \quad \vee \quad top(s) \triangleright top(t))$

$\leadsto$ $[r] = [s] = [t] \quad \wedge \quad top(r) \triangleright top(t)$
because $\triangleright$ is a partial ordering

iii) $[r] = [s] = [t] \quad \wedge \quad top(r) = top(s) = top(t)$

α) $\tau(top(t)) = mult$:

Let $r = f(r_1,\ldots,r_n)$, $s = f(s_1,\ldots,s_n)$, $t = f(t_1,\ldots,t_n)$.

$\leadsto$ $\{r_1,\ldots,r_n\} \gg_{IPOL} \{s_1,\ldots,s_n\} \gg_{IPOL} \{t_1,\ldots,t_n\}$

$\leadsto$ $(\forall j)(\exists i)\ s_i \geq_{IPOL} t_j \quad \wedge \quad (\forall k)(\exists m)\ r_m \geq_{IPOL} s_k$

- $r_m >_{IPOL} s_{k(i)} >_{IPOL} t_j$

$\leadsto$ $r_m >_{IPOL} t_j$
by induction hypothesis

- $r_m >_{IPOL} s_{k(i)} =_{IPOL} t_j \quad \vee$
  $r_m =_{IPOL} s_{k(i)} >_{IPOL} t_j$
  since $s =_{IPOL} r >_{IPOL} t \;\vee\; s >_{IPOL} r =_{IPOL} t$
  implies $s >_{IPOL} t$

- $r_m =_{IPOL} s_{k(i)} =_{IPOL} t_j$

  $\leadsto$ r_m will be removed from the multisets (by definition of $\gg$)

β) $\tau(top(t)) = left$:

Let $r = f(r_1,\dots,r_n)$, $s = f(s_1,\dots,s_n)$, $t = f(t_1,\dots,t_n)$.

$\leadsto$ $(\exists i,j)\ (\forall k < i)\ (\forall m < j)\ r_k =_{IPOL} s_k \;\wedge\; s_m =_{IPOL} t_m \;\wedge\; r_i >_{IPOL} s_i \;\wedge\; s_j >_{IPOL} t_j$

- $i \neq j$

  $\leadsto$ $r_i >_{IPOL} s_i =_{IPOL} t_i \;\vee\; r_j =_{IPOL} s_j >_{IPOL} t_j$

  $\leadsto$ $r_i >_{IPOL} t_i \;\vee\; r_j >_{IPOL} t_j$
  because $s =_{IPOL} r >_{IPOL} t \;\vee\; s >_{IPOL} r =_{IPOL} t$
  implies $s >_{IPOL} t$

  $\leadsto$ $r >_{IPOL} t$
  since all predecessors are equivalent w.r.t $=_{IPOL}$

- $i = j$

  $\leadsto$ $r_i >_{IPOL} s_i >_{IPOL} t_i$

  $\leadsto$ $r_i >_{IPOL} t_i$
  by induction hypothesis

  $\leadsto$ $r >_{IPOL} t$
  analogous with the case $i \neq j$

γ) $\tau(top(t)) = right$:
analogous with β)

C) $>_{IPOL}$ has the subterm property:
We have to show that $\varepsilon \neq u \in O(t) \;\leadsto\; t >_{IPOL} t/u$.
This is valid, since the condition referring to unary operators is required (Note that $[f(x)] >_{IPOL} [x]$ if $[f](x) = x$ must be required, additionally).

D) $>_{IPOL}$ is monotonous, i.e. $t[u \leftarrow r] >_{IPOL} t[u \leftarrow s]$ if $r >_{IPOL} s$:

i) $[r] >_P [s]$

$\leadsto$ $t[u \leftarrow r] >_{IPOL} t[u \leftarrow s]$
since $>_P$ is monotonous

ii) $[r] = [s]$

$\leadsto$ $[t[u \leftarrow r]] = [t[u \leftarrow s]]$

$\leadsto$ the t/u's must be compared because the term structures are identical, otherwise

$\leadsto$ $t[u \leftarrow r] >_{IPOL} t[u \leftarrow s]$
because $r >_{IPOL} s$

E) $>_{IPOL}$ is stable by instantiations:
We must show that $(\forall\sigma, s, t)\ s >_{IPOL} t \leadsto \sigma(s) >_{IPOL} \sigma(t)$. We will prove this fact by induction on $|t|$:

i) $[s] >_P [t]$

$\leadsto$ $\sigma(s) >_{IPOL} \sigma(t)$
since $>_P$ is stable by instantiations

ii) $[s] = [t] \quad \wedge \quad top(s) \triangleright top(t)$

$\leadsto$ $[\sigma(s)] \geq_P [\sigma(t)]$
because $[s] = [t] \leadsto [\sigma(s)] = [\sigma(t)]$

$\leadsto$ $\sigma(s) >_{IPOL} \sigma(t)$
because $(\forall t \in \Gamma)\ top(t) = top(\sigma(t))$

iii) $[s] = [t] \quad \wedge \quad top(s) = top(t)$

Let $s = f(s_1,\ldots,s_n)$, $t = f(t_1,\ldots,t_n)$.

α) $\tau(top(t)) = mult$

$\leadsto$ $\{s_1,\ldots,s_n\} \gg_{IPOL} \{t_1,\ldots,t_n\}$
by definition of $>_{IPOL}$

$\leadsto$ $(\forall j)\ (\exists i)\ s_i \geq_{IPOL} t_j$
by definition of $\gg$

$\rightsquigarrow$ $(\forall j)\ (\exists i)\ \ \sigma(s_i) \geq_{IPOL} \sigma(t_j)$
by induction hypothesis and the fact that
$s =_{IPOL} t \ \rightsquigarrow\ \sigma(s) =_{IPOL} \sigma(t)$

$\rightsquigarrow$ $\{\sigma(s_1),\ldots,\sigma(s_n)\} \gg_{IPOL} \{\sigma(t_1),\ldots,\sigma(t_n)\}$
by definition of $>_{IPOL}$

$\rightsquigarrow$ $\sigma(s) >_{IPOL} \sigma(t)$

β) $\tau(top(t)) = left$

$\rightsquigarrow$ $(\exists i)\ (\forall j < i)\ \ s_j =_{IPOL} t_j \ \ \wedge\ \ s_i >_{IPOL} t_i$

$\rightsquigarrow$ $(\exists i)\ (\forall j < i)\ \ \sigma(s_j) =_{IPOL} \sigma(t_j) \ \ \wedge\ \ \sigma(s_i) >_{IPOL} \sigma(t_i)$
by induction hypothesis and the fact that
$s =_{IPOL} t \ \rightsquigarrow\ \sigma(s) =_{IPOL} \sigma(t)$

$\rightsquigarrow$ $\sigma(s) >_{IPOL} \sigma(t)$
by definition of $>_{IPOL}$

γ) $\tau(top(t)) = right$:
analogous with β) ∎

Proofs of chapter 7

Lemma 7.1.4

Let $>_{IPOL}$ be based on super-linear interpretations, only. Then,

$$s >_{KBOS} t \quad \text{iff} \quad s >_{IPOL} t$$

Proof: Note that the weight of a term $t = f(t_1,...,t_n)$ is defined as

$$\varphi(t) = \varphi(f) + \Sigma\,\varphi(t_i)$$

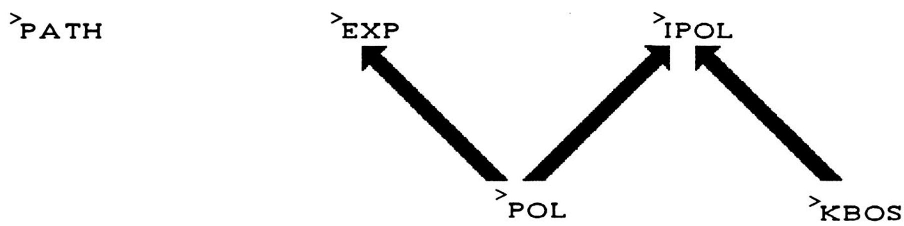
where $\varphi(f)$ is a non-negative real number.

Assume that $[f](x_1,...,x_m) = x_1 + ... + x_m + \varphi(f)$.

⇝ $\varphi(t) = [t]$
since $\varphi(t)$ is the sum of the weights of all function symbols appearing in t

⇝ $s >_{KBOS} t$ iff $s >_{IPOL} t$
because all interpretations are super-linear (which is the precondition) and the structures (w.r.t. the precedence and the recursive call) of both orderings are identical. ■

Lemma 7.2.2



Proof: We have to prove the following relations:

i) $>_{POL} \subset >_{EXP}$: Example 6.3.10

ii) $>_{POL} \subset >_{IPOL}$: Example 6.3.4

iii) $>_{KBOS} \subset >_{IPOL}$: Example 6.3.4

iv) $>_{PATH} \# >_{EXP}$:
- $>_{PATH} \subset >_{EXP}$: Example 7.1.6
- $>_{EXP} \subset >_{PATH}$: Example 6.3.11

v) $>_{PATH} \# >_{IPOL}$:
- $>_{PATH} \subset >_{IPOL}$: Example 7.1.6
- $>_{IPOL} \subset >_{PATH}$: Example 6.3.11

vi) $>_{PATH} \# >_{POL}$: analogous with v)

vii) $>_{PATH} \# >_{KBOS}$:
- $>_{PATH} \subset >_{KBOS}$: $(\neg x \supset y) \vee z \rightarrow (y \vee z) \vee x$
- $>_{KBOS} \subset >_{PATH}$: Example 7.1.2

viii) $>_{EXP} \# >_{IPOL}$:
- $>_{EXP} \subset >_{IPOL}$: Example 6.3.1
- $>_{IPOL} \subset >_{EXP}$: Example 6.3.10

ix) $>_{EXP} \# >_{KBOS}$: analogous with viii)

x) $>_{POL} \# >_{KBOS}$:
- $>_{POL} \subset >_{KBOS}$: Example 6.3.1
- $>_{KBOS} \subset >_{POL}$: Example 2.4.6

Note that $> \# \succ$ denotes the fact that $>$ and $\succ$ are overlapping. ∎

Lemma 7.2.3

The orderings $>_{APO}$, $>_{ACK}$ and $>_{ACP}$ overlap with each other.

Proof: We must show the following relations:

i) $>_{APO} \# >_{ACK}$:
- $>_{APO} \subset >_{ACK}$: $(\neg x \supset y) \vee z \rightarrow \neg(\neg y \wedge \neg z) \vee x$
- $>_{ACK} \subset >_{APO}$: $x \supset (y \vee \mathrm{false}) \rightarrow (x \supset y) \vee x$

with $\wedge, \vee$ are associative-commutative operators

ii) $>_{APO} \# >_{ACP}$: analogous with i)

iii) $>_{ACK} \# >_{ACP}$:
- $>_{ACK} \subset >_{ACP}$: $\mathrm{double}(x) \rightarrow x + x$
- $>_{ACP} \subset >_{ACK}$: $(-x) * x \rightarrow x * (-x)$

with $+$ is an associative-commutative operator whereas $*$ is only associative ∎

Index