Generating Polynomial Orderings for Termination Proofs*

Jürgen Giesl

FB Informatik, Technische Hochschule Darmstadt,
Alexanderstr. 10, 64283 Darmstadt, Germany
Email: giesl@inferenzsysteme.informatik.th-darmstadt.de

Abstract. Most systems for the automation of termination proofs using polynomial orderings are only *semi-automatic*, i.e. the "right" polynomial ordering has to be given by the user. We show that a variation of Lankford's partial derivative technique leads to an easier and slightly more powerful method than most other semi-automatic approaches.
Based on this technique we develop a method for the *automated synthesis* of a suited polynomial ordering.

1 Introduction

One of the most interesting properties of a term rewriting system (trs) is *termination*, cf. [HO80], [DJ90]. A term rewriting system $\mathcal{R}$ is *terminating* for a set of terms $\mathcal{T}$ if there exists no infinite derivation of terms in $\mathcal{T}$. While in general this problem is undecidable [HL78], several methods for proving termination have been presented, cf. [Der87a].

This paper is concerned with the *automation* of termination proofs. Approaches for automated termination proofs using *path orderings* [Pla78], [Der82] are described in [Aït85] and [DF85], an algorithm using *Knuth-Bendix orderings* [KB70] is presented in [Mar87] and a system which is able to handle *general path orderings* is presented in [DH93]. Implementations using *polynomial orderings* [Lan79] have been developed by A. Ben Cherifa and P. Lescanne [BL87] and J. Steinbach [Ste91], [Ste92]. While the systems in [BL87] and [Ste92] only prove termination with a *given* polynomial ordering, Steinbach [Ste91] describes a system that tries to generate a suited polynomial ordering automatically.

In this paper we present a new method for automated termination proofs using polynomial orderings, which is based on a variant of Lankford's partial derivative technique. Our method can be used both in a *semi-automatic* (section 2) and a *fully automated* way (sections 3 and 4). We illustrate its performance and discuss its relation with other approaches for automated termination proofs.

* Technical Report IBN 95/23, Technische Hochschule Darmstadt. This is an extended version of a paper presented at the *6th International Conference on Rewriting Techniques and Applications*, LNCS 914, [Gie95].

2 Semi-Automatic Termination Proofs

The use of *polynomial orderings* for termination proofs has been suggested by D. S. Lankford [Lan79]. A *polynomial interpretation* τ associates an integer multivariate polynomial $f_\tau(x_1, \ldots, x_n)$ with each n-ary function symbol f. To use τ for termination proofs, these polynomials have to be *monotonic* (i.e. $f_\tau(\ldots x \ldots) > f_\tau(\ldots x' \ldots)$ if $x > x'$) and all ground terms have to be mapped into integers that are greater or equal than some lower bound μ (i.e. $\mu \leq \tau(t)$ for all ground terms t). The ordering implicitly defined by a polynomial interpretation τ is called the *corresponding* polynomial ordering $\succ_\tau$ (i.e. $t \succ_\tau s$ iff $\tau(t) > \tau(s)$).

In order to compare *non-ground* terms, τ is extended to interpret variables as variables over the integers. We only regard trs with *finitely* many rules and assume that our signature contains at least one constant, i.e. that there exist ground terms in $\mathcal{T}$. To prove the termination of a trs $\mathcal{R}$, $\mathcal{R}$ has to be *compatible with* a polynomial ordering; i.e. for each rule $l \rightarrow r$ in $\mathcal{R}$, $\tau(l) > \tau(r)$ must hold for all instantiations of the variables. A discussion on the class of term rewriting systems whose termination can be proved using polynomial orderings can be found in [Les86], [Lau88], [HL89], [CL92].

Most systems for "automated" termination proofs using polynomial orderings are semi-automatic, i.e. the user has to provide a polynomial interpretation and the system checks whether the trs is compatible with the corresponding polynomial ordering. In this section we introduce such a semi-automatic system and compare it to the ones in [BL87] and [Ste92].

We will first illustrate our approach with an example from [Bel84] and [BL87]. Let $\mathcal{T}$ consist of all terms constructed from the constant a, the unary function symbol map and the binary function symbol $\circ$. Let $\mathcal{R}$ be the following trs for associativity and endomorphism[1]

$$(x \circ y) \circ z \rightarrow x \circ (y \circ z), \tag{1}$$

$$\mathsf{map}(x) \circ \mathsf{map}(y) \rightarrow \mathsf{map}(x \circ y), \tag{2}$$

$$\mathsf{map}(x) \circ (\mathsf{map}(y) \circ z) \rightarrow \mathsf{map}(x \circ y) \circ z. \tag{3}$$

Our aim is to prove $\mathcal{R}$'s termination by showing that it is compatible with a polynomial ordering $\succ_\tau$. Then the following inequalities have to be true for all instantiations of x, y, z with integers.

$$\circ_\tau\,(\circ_\tau(x, y), z) > \circ_\tau(x, \circ_\tau(y, z)), \tag{4}$$

$$\circ_\tau\,(\mathsf{map}_\tau(x), \mathsf{map}_\tau(y)) > \mathsf{map}_\tau(\circ_\tau(x, y)), \tag{5}$$

$$\circ_\tau\,(\mathsf{map}_\tau(x), \circ_\tau(\mathsf{map}_\tau(y), z)) > \circ_\tau(\mathsf{map}_\tau(\circ_\tau(x, y)), z). \tag{6}$$

[1] This example comes from the area of functional programming. The intended meaning of $\circ$ is composition, variables represent functions and map is a *mapcar*-like operator (whose result is a function).

Let $\succ_\tau$ be the polynomial ordering given by $\mathsf{a}_\tau = 2$, $\mathsf{map}_\tau(x) = 2x$ and $\circ_\tau(x,y) = xy + x$. With this polynomial interpretation inequality (4) is transformed to

$$(xy+x)z + xy + x \;>\; x(yz+z) + x$$

which is equivalent to $xz > 0$. To show that $\mathcal{R}$ is compatible with the polynomial ordering $\succ_\tau$, we therefore have to prove the following inequalities that result from (4) - (6) when using this polynomial ordering (and applying simple arithmetic laws).

$$xz > 0, \tag{7}$$
$$2xy > 0, \tag{8}$$
$$2xyz + 2xy - 2xz > 0. \tag{9}$$

The requirement that $\tau(l) > \tau(r)$ holds for *all* instantiations of the variables with integers is usually too strong. For instance the inequalities (7) - (9) are not valid for all instantiations of x, y, z (e.g. all inequalities are false if $x = 0$).

For a trs $\mathcal{R}$ to be compatible with a polynomial ordering $\succ_\tau$ it is sufficient to demand $\tau(l\sigma) > \tau(r\sigma)$ for all ground substitutions σ [Lan75]. Equivalently, it is sufficient to demand $\tau(l) > \tau(r)$ only for instantiations of variables with those numbers that are *values of ground terms* (i.e. numbers n such that there exists a ground term t with $\tau(t) = n$). In our example ground terms are only mapped into even numbers greater or equal than 2. Therefore it is sufficient if the inequalities (7) - (9) hold for all instantiations of x, y, z with these numbers.

But as in general such a condition is hard to check, a slightly stronger requirement is often used [DJ90]: $\tau(l) > \tau(r)$ is demanded for all instantiations with integers n that are *greater or equal than the minimal value of a ground term* (i.e. numbers n with $n \geq \min\{\tau(t) \mid t \text{ ground term}\}$).

In our example all ground terms t are associated with numbers $\tau(t) \geq 2$. Therefore it is sufficient for $\mathcal{R}$'s termination if the inequalities (7) - (9) hold for all instantiations of x, y, z with integers greater or equal than 2. Now the problem is how to prove such a requirement. Note that in general this question is undecidable [Lan79].

Instead of demanding that inequality (9) should hold for all $x, y, z \geq 2$, it is sufficient if this inequality holds for $x = 2$ and if $2xyz + 2xy - 2xz$ is not decreasing when x is increasing. In other words, the partial derivative of $2xyz + 2xy - 2xz$ with respect to x should be non-negative. Therefore we can replace (9) by the inequalities

$$4yz + 4y - 4z > 0 \quad \text{(resulting from } x = 2\text{)} \quad \text{and} \tag{10}$$
$$2yz + 2y - 2z \geq 0 \quad \text{(resulting from partial derivation).} \tag{11}$$

By further application of this technique (i.e. demanding that (10) and (11) hold for $y = 2$ and that the partial derivatives with respect to y are non-negative) (10) is transformed into the inequalities $4z + 8 > 0$ and $4z + 4 \geq 0$ and (11) is transformed into $2z + 4 \geq 0$ and $2z + 2 \geq 0$. Finally, the variable z is eliminated in the same way. This yields the inequalities $16 > 0$, $4 \geq 0$, $12 \geq 0$, $4 \geq 0$, $8 \geq 0$,

$2 \geq 0$, $6 \geq 0$ and $2 \geq 0$. As these resulting inequalities (between *numbers*) are true, the original inequality (9) also holds for all $x, y, z \geq 2$. The validity of the other two inequalities (7) and (8) can be proved in the same way.

A semi-automatic system for termination proofs using polynomial orderings mainly consists of a procedure to check whether a polynomial $p(x_1, \ldots, x_n)$ is positive for all instantiations of the variables $x_1, \ldots, x_n$ with integers greater or equal than a minimal value $\mu \in \mathbb{Z}$. In the above example we have used the following two *differentiation rules* to prove such requirements.

$$\frac{p(\ldots x \ldots) > 0}{p(\ldots \mu \ldots) > 0, \ \frac{\partial p(\ldots x \ldots)}{\partial x} \geq 0} \qquad \text{(Diff1)}$$

$$\frac{p(\ldots x \ldots) \geq 0}{p(\ldots \mu \ldots) \geq 0, \ \frac{\partial p(\ldots x \ldots)}{\partial x} \geq 0} \qquad \text{(Diff2)}$$

By repeated application of the differentiation rules (Diff1) and (Diff2) polynomial inequalities of the form $p(x_1, \ldots, x_n) > 0$ are transformed into inequalities between *numbers* (i.e. $n > 0$ or $n \geq 0$, where $n \in \mathbb{Z}$). This transformation is *sound*, i.e. if the resulting inequalities between numbers are true, then $p(x_1, \ldots, x_n) > 0$ holds for all $x_1, \ldots, x_n \geq \mu$.

This results in the following method for semi-automatic termination proofs where the polynomial interpretation τ and a minimal value $\mu \leq \min\{\tau(t) \mid t$ ground term$\}$ have to be provided by the user.

Theorem 1 (Semi-Automatic Termination Proofs). *Let $\mathcal{R}$ be a trs, let τ be a monotonic polynomial interpretation mapping all ground terms into integers that are greater or equal then μ. Repeated application of the differentiation rules (Diff1) and (Diff2) to $\tau(l) - \tau(r)$ (for all rules $l \to r$ in $\mathcal{R}$) yields a unique set of inequalities between numbers (i.e. $n > 0$ or $n \geq 0$). If these inequalities are true, then $\mathcal{R}$ is terminating.*

The differentiation rules (Diff1) and (Diff2) are based on the partial derivative method of Lankford [Lan76]. But Lankford's method can only prove that a polynomial is *eventually positive* (i.e. $p(x_1, \ldots, x_n) > 0$ holds for large enough x_i). Note that it is not sufficient for the termination of a trs $\mathcal{R}$ if there exists a polynomial interpretation τ such that $\tau(l) - \tau(r)$ is eventually positive for each rule $l \to r$ in $\mathcal{R}$ [Der87b]. For instance, the trs with the rule $x \to \mathsf{a}$ is not terminating although $\tau(x) - \tau(\mathsf{a})$ is eventually positive for every polynomial interpretation τ.

A different semi-automatic method has been presented by Ben Cherifa and Lescanne [BL87]. As their approach uses a heuristic (which can fail), their method was improved by Steinbach [Ste92] who developed a system which has the same power as the one of [BL87] if the latter is extended by a backtracking component. Moreover, he eliminated the restriction of [BL87] to the fixed minimal value $\mu = 2$. A comparison of our method with the ones of [Ste92] and [BL87] (the latter one extended by backtracking and arbitrary μ) leads to the following results[2]:

[2] Detailed proofs of the following observations can be found in the appendix.

- *If [Ste92] and [BL87] can prove a polynomial p positive, then our method can do so as well.*
  The reason is, that if $p > 0$ can be proved by the methods of [Ste92] and [BL87] then $\frac{\partial p}{\partial x} \geq 0$ can also be proved with their methods (and therefore it must be valid). Hence, repeated application of (Diff1) and (Diff2) transforms $p > 0$ into a set of valid inequalities.
- *If our method can prove p positive for all $x_1, \ldots, x_n \geq \mu$, then there exists a $\mu' \geq \mu$ such that the methods of [Ste92] and [BL87] can prove p positive for all $x_1, \ldots, x_n \geq \mu'$.*
  But it is not always possible to choose $\mu' = \mu$. For example, the systems of [Ste92] and [BL87] can prove $x^2 - 2x + 2 > 0$ only for $x \geq 3$ while our method can already prove it for $x \geq 1$.
- *While the worst case complexity of the systems in [Ste92] and [BL87] is exponential in the number of monomials in p, our method is exponential in the number of its variables.*
  More precisely, the complexity of our method to prove a r-variate polynomial with degree d positive is $O(d^r)$.

In [Ste92], Steinbach also suggested the additional use of the arithmetic-mean-geometric-mean inequality. This allows proofs that are not possible with our approach. But as we know of no heuristic for deciding when to apply this inequality and when to prove $p > 0$ in the usual way, we do not integrate it into our method.

3 A Termination Criterion with Variable Coefficients

Now our aim is to develop a method for the *automated generation* of polynomial orderings. For each trs $\mathcal{R}$ we have to synthesize a polynomial ordering $\succ_\tau$ such that $\mathcal{R}$ is compatible with $\succ_\tau$. For that purpose every function symbol has to be associated with a polynomial. To determine the coefficients of these polynomials we proceed in two steps (which will be described in this and the next section respectively):

Given a trs $\mathcal{R}$ we first compute *constraints* for the choice of the coefficients. If the coefficients satisfy these constraints, then $\mathcal{R}$ is compatible with the corresponding polynomial ordering. Therefore the second step consists of the generation of coefficients satisfying these constraints.

In the last section we presented a termination criterion using associations with polynomials whose coefficients are *integer numbers*. We will now extend theorem 1 to a termination criterion using polynomials with *variable coefficients*. This yields a method to generate a set of inequalities representing constraints for the instantiation of the coefficients. Every instantiation satisfying these constraints is compatible with the given trs. A discussion on how to find such an instantiation automatically follows in section 4.

To generate a polynomial interpretation we first have to decide on the *maximum degree* of the polynomials. We follow a heuristic from [Ste91] and associate a *simple-mixed*[3] polynomial with each function symbol. Alternatively, one could first try to prove the termination of $\mathcal{R}$ with polynomials of maximum degree 1, then try polynomials with maximum degree 2 etc. and give up if the maximum degree exceeds a certain upper bound.

If we attempt a termination proof with simple-mixed polynomials, in our example the constant a is associated with a number a_0, the unary function symbol map is associated with a polynomial $\mathsf{map}_\tau(x) = m_0 + m_1 x$ (or $m_0 + m_2 x^2$) and $\circ$ is associated with $\circ_\tau(x, y) = c_0 + c_1 x + c_2 y + c_3 xy$. Here we use a polynomial interpretation τ which maps function symbols to polynomials with *variable coefficients* $a_0, m_0, m_1, c_0, c_1, c_2, c_3$.

Now we have to find an instantiation of the variable coefficients $a_0, \ldots, c_3$ such that $\mathcal{R}$ is compatible with the corresponding polynomial ordering, i.e. $\tau(l) - \tau(r) > 0$ should hold for each rule $l \to r$ in $\mathcal{R}$. Therefore in our example we have to generate an instantiation of $a_0, \ldots, c_3$ such that the inequalities (4) - (6) are true for all instantiations of the *rule variables* x, y, z with integers greater or equal than the minimal value of a ground term.

Using the above polynomial interpretation with variable coefficients (and applying simple arithmetic laws) inequality (4) becomes

$$c_0c_1 - c_0c_2 + (c_1^2 - c_1 - c_0c_3)\, x + (c_2 - c_2^2 + c_0c_3)\, z + (c_1c_3 - c_2c_3)\, xz \;>\; 0. \quad (12)$$

The problem is that we cannot directly check whether an instantiation of the variable coefficients $c_0, \ldots, c_3$ makes this inequality valid *for all* $x, z \geq \min\{\tau(t) \mid t \text{ ground term}\}$. Therefore we will transform (12) into new inequalities which *do not contain the rule variables* x and z any more. Then for each instantiation of the variable coefficients it is trivial to check whether they satisfy these new inequalities. The invariant of this transformation is that every instantiation of $c_0, \ldots, c_3$ satisfying the new inequalities also satisfies the original inequalities *for all* $x, z \geq \min\{\tau(t) \mid t \text{ ground term}\}$. For the transformation we will again make use of the termination criterion of theorem 1 and the differentiation rules presented in the last section.

Let μ be a new variable and let us assume for the moment that μ is instantiated with a value less or equal than $\min\{\tau(t) \mid t \text{ ground term}\}$. Then we can apply the differentiation rules (Diff1) and (Diff2) to transform (12) into inequalities without the variables x and z. We obtain

$$c_0c_1 - c_0c_2 + (c_1^2 - c_1 + c_2 - c_2^2)\mu + (c_1c_3 - c_2c_3)\mu^2 > 0, \quad (13)$$
$$c_1^2 - c_1 - c_0c_3 + (c_1c_3 - c_2c_3)\mu \geq 0, \quad (14)$$
$$c_2 - c_2^2 + c_0c_3 + (c_1c_3 - c_2c_3)\mu \geq 0, \quad (15)$$
$$c_1c_3 - c_2c_3 \geq 0. \quad (16)$$

[3] A non-unary polynomial p is *simple-mixed* iff all its exponents are not greater than 1 (i.e. $2xy - 5xyz$ is simple-mixed while $3x^2y$ is not). A unary polynomial p is simple-mixed if it has the form $\alpha_0 + \alpha_1 x$ or $\alpha_0 + \alpha_2 x^2$. In [Ste91] Steinbach conducted 320 experiments with trs from literature and noticed that 96 % of those trs which are compatible with a polynomial ordering are compatible with a *simple-mixed* polynomial ordering.

If we find an instantiation of the variable coefficients such that (13) - (16) are valid, then due to theorem 1 inequality (12) holds for all $x, z \geq \mu$.

We proceed analogously for (5) and (6) and obtain inequalities which only contain the variable coefficients $a_0, \ldots, c_3$ and μ, but not the rule variables x, y, z. If there exists an instantiation of the variable coefficients satisfying the resulting inequalities, this instantiation also satisfies the original inequalities (4) - (6) for all $x, y, z \geq \mu$.

In other words, the obtained inequalities are constraints for the instantiation of the variable coefficients. If an instantiation satisfies these constraints, then the trs $\mathcal{R}$ is compatible with the corresponding polynomial ordering.

But to imply $\mathcal{R}$'s termination we furthermore have to ensure that all polynomials f_τ are *monotonic*. Therefore we also have to demand

$$(x > x') \;\rightarrow\; (\mathsf{map}_\tau(x) > \mathsf{map}_\tau(x')), \tag{17}$$

$$(x > x') \;\rightarrow\; (\circ_\tau(x, y) > \circ_\tau(x', y)), \tag{18}$$

$$(y > y') \;\rightarrow\; (\circ_\tau(x, y) > \circ_\tau(x, y')). \tag{19}$$

Instead of demanding these conditions for all integers x, y, x', y' it is again sufficient if the inequalities hold for all $x, y, x', y' \geq \mu$. Our aim is to obtain constraints for the instantiation of the variable coefficients that are sufficient for the validity of (17) - (19). Therefore we have to transform these inequalities into inequalities without the variables x, y, x', y'. While in inequality (12) the rule variables x and z could be eliminated by application of the differentiation rules, this is not possible for the monotonicity inequalities (17) - (19).

The reason is that direct application of the differentiation rules is not sound for *conditional* inequalities. Assume, for instance, we want to check whether

$$(y - 1 \geq 0) \;\rightarrow\; (y - 2 \geq 0) \tag{20}$$

holds for all $y \geq 0$. This is true for $y = 0$ and the partial derivatives of $y - 1$ and $y - 2$ are both positive. Nevertheless, the instantiation $y = 1$ falsifies formula (20). Therefore the differentiation technique of the last section only works for unconditional inequalities.

To be able to eliminate the rule variables with the differentiation rules the monotonicity inequalities (17) - (19) have to be transformed into *unconditional* inequalities.

Inequality (18) guarantees that if x is increasing, $\circ_\tau(x, y)$ is also increasing. Instead of (18) we can therefore demand that the partial derivative of $\circ_\tau(x, y)$ with respect to x is positive. In our example we have $\frac{\partial \circ_\tau(x,y)}{\partial x} = c_1 + c_3 y$ and therefore (18) is replaced by $c_1 + c_3 y > 0$. As this is an unconditional inequality, we can now use (Diff1) again to eliminate the remaining rule variable y. In this way we replace (17) - (19) by

$$m_1 > 0, \tag{21}$$

$$c_1 + c_3 y > 0, \tag{22}$$

$$c_2 + c_3 x > 0. \tag{23}$$

So in general instead of $(x > x') \rightarrow (f_\tau(\ldots x \ldots) > f_\tau(\ldots x' \ldots))$ we will always demand

$$\frac{\partial f_\tau(\ldots x \ldots)}{\partial x} > 0.$$

Still we have to ensure that the variable μ is really instantiated with a value less or equal than the minimal value of a ground term. For that purpose we demand

$$c_\tau - \mu \geq 0$$

for all constants c of the signature. For non-constant function symbols f we demand that application of f to μ yields a value greater or equal than μ, i.e.

$$f(\mu, \ldots, \mu) - \mu \geq 0.$$

This condition is sufficient for the requirement $\mu \leq \min\{\tau(t) \mid t \text{ ground term}\}$. The reason is that as each function symbol f is associated with a *monotonic* polynomial f_τ, the inequality $f_\tau(x_1, \ldots, x_n) \geq \mu$ holds for all $x_1, \ldots, x_n \geq \mu$. Therefore in our example the instantiation of the variables also has to satisfy the inequalities

$$a_0 - \mu \geq 0, \tag{24}$$

$$m_0 + m_1\mu - \mu \geq 0, \tag{25}$$

$$c_0 + c_1\mu + c_2\mu + c_3\mu^2 - \mu \geq 0. \tag{26}$$

The following theorem summarizes our termination criterion using polynomial interpretations with variable coefficients.

Theorem 2 (Termination Criterion with Variable Coefficients). *Let $\mathcal{R}$ be a trs, let τ be a polynomial interpretation with variable coefficients. Repeated application of the differentiation rules (Diff1) and (Diff2) to*

$$\begin{array}{rl} \tau(l) - \tau(r) > 0 & \textit{for all rules } l \rightarrow r \textit{ in } \mathcal{R}, \\ \frac{\partial f_\tau(\ldots x \ldots)}{\partial x} > 0 & \textit{for all function symbols } f, \\ f_\tau(\mu, \ldots, \mu) - \mu \geq 0 & \textit{for all function symbols } f, \\ c_\tau - \mu \geq 0 & \textit{for all constants } c \end{array}$$

yields a unique set of inequalities containing no rule variables any more. If there exists an instantiation of the variable coefficients and the variable μ satisfying the resulting inequalities, then $\mathcal{R}$ is terminating.

So in our example we start with the rule inequalities (4) - (6), the monotonicity inequalities (21) - (23) and the inequalities (24) - (26) which ensure correct instantiation of μ. Subsequently the rule variables x, y, z are eliminated by repeated application of the differentiation rules (Diff1) and (Diff2).

The resulting inequalities are satisfied by the instantiation corresponding to the polynomial interpretation given in section 2 (i.e. $\mu = 2$, $a_0 = 2$, $m_0 = 0$,

$m_1 = 2$, $c_0 = 0$, $c_1 = 1$, $c_2 = 0$, $c_3 = 1$). Therefore by theorem 2 the termination of $\mathcal{R}$ is proved.

Instead of the differentiation rules we could also use Steinbach's technique [Ste92] for the elimination of the rule variables x, y, z (as suggested in [Ste91]). But while Steinbach's technique introduces *several* new variables, the advantage of (Diff1) and (Diff2) is that these rules introduce only *one* new variable μ. For the generation of a polynomial ordering compatible with $\mathcal{R}$ we therefore only have to find an instantiation of the variable coefficients and μ.

In the next section we discuss how to find such an instantiation automatically.

4 A Fully Automated Termination Proof Procedure

In theorem 2 we introduced a method to automatically generate a set of inequalities only containing variable coefficients and the variable μ. These inequalities represent constraints for the instantiation of the variable coefficients. To prove the termination of a trs $\mathcal{R}$ mechanically we now have to synthesize an instantiation of these variables satisfying the inequalities.

When examining term rewriting systems occurring in the literature we noticed that most termination proofs with polynomial interpretations only use polynomials whose coefficients are 0, 1 or 2. Checking whether a certain instantiation of variables with numbers satisfies the inequalities resulting from theorem 2 can be done very efficiently. Therefore we suggest to apply a "generate and test" approach first which generates all instantiations of the variables with numbers from $\{0, 1, 2\}$ until one of these instantiations satisfies the inequalities. This results in a fully automated termination proof procedure which succeeds for most of those term rewriting systems which are compatible with a polynomial ordering.

Nevertheless there do exist term rewriting systems which require a polynomial ordering with coefficients other than 0, 1 or 2. A trivial example is the trs $\mathsf{a} \rightarrow \mathsf{b}$, $\mathsf{b} \rightarrow \mathsf{c}$, $\mathsf{c} \rightarrow \mathsf{d}$ where $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}$ are constants. It is only compatible with polynomial orderings that use at least four different coefficients (e.g. $\mathsf{a}_\tau = 3, \mathsf{b}_\tau = 2, \mathsf{c}_\tau = 1, \mathsf{d}_\tau = 0$).

It is undecidable whether there exists an instantiation with *integers* satisfying a set of inequalities. But if we regard instantiations with *real* numbers this problem becomes decidable [Tar51]. Then decision methods for elementary algebra (e.g. [Tar51], [Coh69], [Col75]) can be used for the synthesis of the "right" instantiation.

To be compatible with an *integer* polynomial ordering is sufficient for the termination of a trs. This is because for a non-terminating trs there would have to be an infinite derivation of ground terms. But as every ground term is mapped to an integer greater or equal than a lower bound μ, there would have to be a bounded infinite descending chain of integers which leads to a contradiction.

Unfortunately, this method for termination proofs does not work for *real* polynomial orderings. The reason is that there exist bounded infinite descending chains of reals (e.g. $1, \frac{1}{2}, \frac{1}{4}, \ldots$) as the distance between two different real numbers can be infinitesimally small. Therefore e.g. the trs $\mathsf{a} \to \mathsf{g}(\mathsf{a})$ is compatible with the polynomial ordering defined by $\mathsf{a}_\tau = 1$ and $\mathsf{g}_\tau(x) = \frac{1}{2}x$ although it is not terminating.

Consequently, the termination criterion of theorem 2 becomes unsound if instantiations with real numbers are allowed. If a is associated with a_0 and g is associated with $g_0 + g_1 x$ then the method of theorem 2 constructs inequalities that are satisfied by the instantiation $\mu = 0, a_0 = 1, g_0 = 0, g_1 = \frac{1}{2}$. So satisfiability of these inequalities by an instantiation with real numbers is not sufficient for termination.

In the following we will therefore develop a refined termination criterion which can also be used for instantiations with real numbers. This enables the application of decision methods for elementary algebra to generate a suited instantiation of the variable coefficients.

In [Der79] Dershowitz proposed a method for proving termination using *real* polynomial orderings. He showed that it is sufficient for termination if a trs is compatible with a *simplification ordering* (i.e. a monotonic ordering $\succ$ possessing the *subterm property* $f(\ldots x \ldots) \succ x$). For a survey on simplification orderings see [Ste89], [Ste93] and [Ste94]. This result was strengthened in [Der82] by stating that for termination it is already sufficient if the trs is compatible with the strict part $\succ$ of a *quasi-simplification ordering* $\succeq$. A quasi-simplification ordering $\succeq$ is a *quasi*-ordering (i.e. transitive and reflexive) that is monotonic (i.e. $f(\ldots x \ldots) \succeq f(\ldots x' \ldots)$ if $x \succeq x'$) and possesses the subterm property (i.e. $f(\ldots x \ldots) \succeq x$).

Every polynomial interpretation τ defines a corresponding polynomial *quasi*-ordering $\succeq_\tau$ (i.e. $t \succeq_\tau s$ iff $\tau(t) \geq \tau(s)$). As suggested in [Der82], we prove the termination of a trs $\mathcal{R}$ by showing that it is compatible with a (possibly *real*) polynomial ordering $\succ_\tau$ whose corresponding *quasi*-ordering $\succeq_\tau$ is a quasi-simplification ordering. In other words, we have to ensure *non-strict* monotonicity and the *non-strict* subterm property.

Remember that in section 3 we had to guarantee that the instantiation of the variable coefficients resulted in a *monotonic* polynomial interpretation. For that purpose the instantiation had to satisfy the inequality $(x > x') \to (f_\tau(\ldots x \ldots) > f_\tau(\ldots x' \ldots))$ for all function symbols f.

Now instead of this requirement we have to guarantee the monotonicity and the subterm property of the corresponding polynomial quasi-ordering. Therefore in our example instead of (17) - (19) we have to demand

$$(x \geq x') \;\to\; (\mathsf{map}_\tau(x) \geq \mathsf{map}_\tau(x')), \tag{27}$$

$$(x \geq x') \;\to\; (\circ_\tau(x, y) \geq \circ_\tau(x', y)), \tag{28}$$

$$(y \geq y') \;\to\; (\circ_\tau(x, y) \geq \circ_\tau(x, y')) \quad \text{and} \tag{29}$$

$$\mathsf{map}_\tau(x) - x \geq 0, \tag{30}$$

$$\circ_\tau(x, y) - x \geq 0, \tag{31}$$

$$\circ_\tau(x, y) - y \geq 0. \tag{32}$$

Before applying the differentiation rules we again have to transform the monotonicity formulas (27) - (29) into unconditional inequalities. In contrast to (17) - (19) the above formulas only demand a *non-strict* monotonicity, i.e. if the arguments are increasing then the result of the polynomials should not be decreasing. Instead of demanding that the partial derivatives of the polynomials should be *positive* as in the last section it is therefore now sufficient to demand that they are *non-negative*. So we replace (27) - (29) by

$$m_1 \geq 0, \tag{33}$$

$$c_1 + c_3 y \geq 0, \tag{34}$$

$$c_1 + c_2 x \geq 0. \tag{35}$$

To ensure correct instantiation of the variable μ we demand that $\mu \leq c_\tau$ holds for all constants c. This condition is equivalent to $\mu \leq \min\{\tau(t) \mid t \text{ ground term}\}$ because of the *subterm property*.

Summing up, we obtain the following alternative termination criterion which also allows instantiations with real numbers.

Theorem 3 (Termination Criterion with Real Variable Coefficients). *Let $\mathcal{R}$ be a trs, let τ be a polynomial interpretation with variable coefficients. Repeated application of the differentiation rules (Diff1) and (Diff2) to*

$$\begin{array}{rl} \tau(l) - \tau(r) > 0 & \textit{for all rules } l \to r \textit{ in } \mathcal{R}, \\ \frac{\partial f_\tau(\ldots x \ldots)}{\partial x} \geq 0 & \textit{for all function symbols } f, \\ f_\tau(\ldots x \ldots) - x \geq 0 & \textit{for all function symbols } f, \\ c_\tau - \mu \geq 0 & \textit{for all constants } c \end{array}$$

yields a unique set of inequalities containing no rule variables any more. If there exists an instantiation of the variable coefficients and the variable μ with real numbers which satisfies the resulting inequalities, then $\mathcal{R}$ is terminating.

So in our example we begin with the rule inequalities (4) - (6), the non-strict monotonicity inequalities (33) - (35), the inequalities (30) - (32) guaranteeing the subterm property and inequality (24) which ensures correct instantiation of μ. Then we eliminate the rule variables x, y, z by the differentiation rules (Diff1) and (Diff2). The resulting inequalities are again satisfied by the instantiation mentioned in section 3.

In contrast to theorem 2 the above theorem is also sound for instantiations with *real* numbers. While for the non-terminating trs $\mathsf{a} \to \mathsf{g}(\mathsf{a})$ theorem 2 constructed inequalities that are satisfied by a real instantiation, the inequalities resulting from theorem 3 are unsatisfiable. The reason is that $\mathsf{a}_\tau > \mathsf{g}_\tau(\mathsf{a}_\tau)$ is a contradiction to the subterm property.

Using the criterion of theorem 3 we can now apply decision methods for elementary algebra to determine whether there exists a (real) instantiation of the variable coefficients satisfying the resulting inequalities. If such an instantiation exists, then the trs is terminating.

For instance, in our example one of the inequalities resulting from the first rule (1) is

$$c_1 c_3 - c_2 c_3 \ \geq \ 0. \tag{16}$$

The left hand side of this inequality is a polynomial $c_1c_3 - c_2c_3$ whose *variables* c_1, c_2, c_3 are variable coefficients of the polynomial interpretation. The *coefficients* of this polynomial are 1 and -1.

So in general, to apply the termination criterion of theorem 3 we have to find an instantiation satisfying a set of (strict and non-strict) inequalities

$$p_1 > 0, \ldots, p_k \geq 0 \tag{36}$$

where the polynomials $p_1, \ldots, p_k$ are polynomials whose variables are the variable coefficients of the polynomial interpretation and μ and whose coefficients are integers. In other words, we have to check the validity of an existential formula of the following form (where $c_1, \ldots, c_m$ denote the variable coefficients and μ).

$$\exists c_1, \ldots, c_m \quad p_1 > 0 \wedge \ldots \wedge p_k \geq 0. \tag{37}$$

A decision method for elementary algebra decides whether such a formula is valid over the real numbers. The most efficient known decision method for elementary algebra is the *cylindrical algebraic decomposition* algorithm by G. E. Collins ([Col75], [ACM84], [Hon92]). To prove the validity of a formula of the form (37), Collins' algorithm computes a partition of $\mathbb{R}^m$ into a finite number of subsets (a so-called <u>c</u>ylindrical <u>a</u>lgebraic <u>d</u>ecomposition) such that the polynomials $p_1, \ldots, p_k$ do not change their sign in these subsets (i.e. they are either positive, negative or equal to zero for all instantiations of $c_1, \ldots, c_m$ with numbers from one subset). For example, a cad for the formula $\exists c \; c^2 - 2 > 0$ is

$$(-\infty, -\sqrt{2}),\ \{-\sqrt{2}\},\ (-\sqrt{2}, \sqrt{2}),\ \{\sqrt{2}\},\ (\sqrt{2}, \infty). \tag{38}$$

As the polynomials in $p_1, \ldots, p_k$ do not change their sign in the subsets of the cad, the validity of the inequalities (36) over such a subset can be determined by checking their validity at *one* arbitrary *sample point* belonging to that subset. The validity of the inequalities (36) over *all* subsets can therefore be determined by checking them at *finitely* many sample points. This can be done effectively.

As the cad covers the whole $\mathbb{R}^m$, we can determine the validity of formula (37) by checking the validity of the inequalities (36) over each subset of the cad. For that purpose we choose a sample point out of each subset. As (37) is an *existential* formula, it is valid iff the inequalities (36) are valid at *one* of the sample points.

For the cad (38) we can choose the sample points $\{-2, -\sqrt{2}, 0, \sqrt{2}, 2\}$. As $c^2 - 2 > 0$ is valid at both the sample points -2 and 2, the validity of $\exists c \; c^2 - 2 > 0$ is proved.

The disadvantage of this (complete) approach is that all known decision methods for elementary algebra are very time-consuming. For that reason these methods have been rarely used for automated termination proofs.

Therefore we suggest an *incomplete*, more efficient modification of Collins' algorithm, which is adapted to the specific problem of termination proofs. As we know of no trs whose termination proof requires a polynomial interpretation with non-rational real coefficients, we have restricted the algorithm to *rational* instead of *real* numbers. This eases the implementation of the algorithm considerably and we avoid a disadvantage mentioned in [Der87a], i.e. the generated (rational) polynomial interpretation can be directly printed out to the user which would not be possible if we used real numbers. Moreover, we have introduced execution time limits for each step of Collins' algorithm. If the time limit for the actual step is exceeded, then the algorithm can only use the results of the actual step computed so far and has to carry on with the next step.

The main effect of these modifications is that *not all* of the sample points are computed any more. Then we can no longer check the validity of the inequalities (36) over *all* subsets of the cad, but only over *some* of them.

As mentioned above, it is sufficient for the validity of an *existential* formula (37) if the inequalities (36) are at least valid at one sample point. If we do not check the validity at all sample points, there may be a point where the inequalities are valid, but we possibly do not find it. Therefore the proof for a valid formula (37) can fail, i.e. the method becomes *incomplete*. But if we find a sample point where the inequalities (36) are valid, then the validity of formula (37) is in fact proved, i.e. the method remains *sound*. So we use Collins' algorithm not as a decision method, but only as a *heuristic*.

Note that in contrast to Collins' original algorithm this modified algorithm is not sound for non-existential formulas. So the modification can only be used because for our termination criterion from theorem 3 we just have to determine the validity of an *existential* formula. Therefore the elimination of the rule variables (by the differentiation rules (Diff1) and (Diff2)) is absolutely necessary to enable the use of the modified cad-algorithm.

To sum up, we propose the following termination proof procedure:

1. Construct a set of inequalities $p_1 > 0, \ldots, p_k \geq 0$ as described in theorem 3 (using a polynomial interpretation with possibly variable coefficients).
2. Check whether these inequalities are satisfied by an instantiation with numbers from $\{0, 1, 2\}$.
3. If not, try to prove their satisfiability by a modified version of Collins' algorithm.

An alternative approach for the automated generation of the "right" polynomial interpretation has been presented by Steinbach [Ste91]. It is based on a technique of approximating polynomials by monomials which can be useful if the number of variable coefficients is small. In these cases Steinbach's method may also be used to search for an instantiation that satisfies the inequalities resulting from theorem 3.

We have implemented our termination proof procedure in Common Lisp on a Sun SPARC-2. Table 1 illustrates its performance with some examples. The second row contains the execution time our algorithm needs to generate a polynomial interpretation which is compatible with the trs in the first row.

Example	Time
Nested Function Symbols ([Ste91, Example 8.1])	0.1 sec.
Endomorphism & Associativity ([Bel84], [BL87])	0.1 sec.
Running Example 6.1 in [Ste91] (by A. Middeldorp)	0.2 sec.
Binomial Coefficients ([Ste91, Example 8.8], [Ste92, Example 13])	1.6 sec.
Distributivity & Associativity ([Der87a, p. 78])	1.9 sec.

Table 1. Performance of our method.

5 Conclusion

We have presented an efficient, powerful and easy to implement algorithm for termination proofs using polynomial orderings which can be used both in a semi-automatic and in a fully automated way. Our method has been implemented[4] and proved successful (cf. table 1).

Acknowledgements
I would like to thank Jürgen Brauburger, Caroline Claus, Alexander Friedl, Stefan Gerberding, Thomas Kolbe, Jens Marschner, Martin Protzen and Christoph Walther for comments and support.

A Proofs

In this appendix we give detailed proofs for the observations in section 2 concerning the comparison of our method with the ones of [Ste92] and [BL87].

Observation 1 *If [Ste92] and [BL87] can prove a polynomial p positive, then our method can do so as well.*

Proof:
It is sufficient to prove the following conjecture:

$$\begin{array}{c}\text{If } p > 0 \text{ (or } p \geq 0\text{) can be proved with the method of Steinbach,}\\ \text{then } \frac{\partial p}{\partial x_i} \geq 0 \text{ can also be proved with his method.}\end{array} \tag{39}$$

From this conjecture we can conclude the original observation, because due to the *soundness* of Steinbach's method, (39) implies that repeated application of the differentiation rules yields *valid* inequalities between numbers. (Here "$p > 0$" abbreviates "$p(x_1, \ldots, x_r) > 0$ for all $x_1, \ldots, x_r \geq \mu$".)

[4] The implementation is available by anonymous ftp from `kirmes.inferenzsysteme.informatik.th-darmstadt.de` under `pub/termination`.

Let $p = \sum \alpha_{k_1,\dots,k_r} x_1^{k_1} \dots x_r^{k_r} - \sum \beta_{i_1,\dots,i_r} x_1^{i_1} \dots x_r^{i_r}$ be a polynomial, where $\alpha_{k_1,\dots,k_r}$ and $\beta_{i_1,\dots,i_r}$ are *positive* numbers. If the inequality $p \geq 0$ can be proved with Steinbach's method, then the following inequalities (featuring new variables $y_{k_1,\dots,k_r,i_1,\dots,i_r}$) must be satisfiable.

for all $k_1, \dots, k_r$:

$$\alpha_{k_1,\dots,k_r} \succeq \sum_{k_1 \geq i_1,\dots,k_r \geq i_r} y_{k_1,\dots,k_r,i_1,\dots,i_r}, \tag{40}$$

for all $i_1, \dots, i_r$:

$$\sum_{k_1 \geq i_1,\dots,k_r \geq i_r} \mu^{(\sum(k_j - i_j))} \; y_{k_1,\dots,k_r,i_1,\dots,i_r} \succeq \beta_{i_1,\dots,i_r}. \tag{41}$$

Let $y_{k_1,\dots,k_r,i_1,\dots,i_r} := \gamma_{k_1,\dots,k_r,i_1,\dots,i_r}$ be an instantiation (with non-negative real numbers) satisfying the inequalities (40) and (41).

We will only prove (39) for the partial derivative with respect to x_1. The partial derivative of p is

$$\frac{\partial p}{\partial x_1} = \sum_{k_1 \neq 0} k_1 \; \alpha_{k_1,\dots,k_r} \; x_1^{k_1-1} x_2^{k_2} \dots x_r^{k_r} - \sum_{i_1 \neq 0} i_1 \; \beta_{i_1,\dots,i_r} \; x_1^{i_1-1} x_2^{i_2} \dots x_r^{i_r}.$$

Steinbach's method can prove the inequality $\frac{\partial p}{\partial x_1} \geq 0$ if the following inequalities (featuring the new variables $z_{k_1,\dots,k_r,i_1,\dots,i_r}$) are satisfiable.

for all $k_1, \dots, k_r$, where $k_1 \neq 0$:

$$k_1 \; \alpha_{k_1,\dots,k_r} \succeq \sum_{k_1 \geq i_1, k_2 \geq i_2 \dots, k_r \geq i_r, \, i_1 \neq 0} z_{k_1-1,k_2\dots,k_r,i_1-1,\dots,i_r}, \tag{42}$$

for all $i_1, \dots, i_r$, where $i_1 \neq 0$:

$$\sum_{k_1 \geq i_1, k_2 \geq i_2, \dots, k_r \geq i_r} \mu^{(\sum(k_j - i_j))} \; z_{k_1-1,k_2,\dots,k_r,i_1-1,i_2,\dots,i_r} \succeq i_1 \; \beta_{i_1,\dots,i_r}. \tag{43}$$

Consider the instantiation $z_{k_1-1,k_2\dots,k_r,i_1-1,i_2,\dots,i_r} := k_1 \; \gamma_{k_1,\dots,k_r,i_1,\dots,i_r}$. Then (40) implies (42) (as all $\gamma_{\dots}$ are non-negative). Moreover, (41) implies (43) (as $k_1 \geq i_1$). Hence, (39) is proved.

Observation 2 *If our method can prove p positive for all $x_1, \dots, x_n \geq \mu$, then there exists a $\mu' \geq \mu$ such that the methods of [Ste92] and [BL87] can prove p positive for all $x_1, \dots, x_n \geq \mu'$.*

Proof:
Let p be a polynomial as above. If $p > 0$ (or $p \geq 0$) can be proved by application of the differentiation rules, then for each $\beta_{i_1,\dots,i_r}$ there must be an $\alpha_{k_1,\dots,k_r} \geq \beta_{i_1,\dots,i_r}$ (such that $k_1 \geq i_1, \dots, k_r \geq i_r$). Therefore if μ' is sufficiently large, then $p > 0$ (or $p \geq 0$) can also be proved using the method of Steinbach.

Observation 3 *While the worst case complexity of the systems in [Ste92] and [BL87] is exponential in the number of monomials in p, our method is exponential in the number of its variables.*

Proof:
For a proof on the complexity of the systems in [Ste92] and [BL87] the reader is referred to [Ste92]. The complexity of our method to prove $p > 0$ (or $p \geq 0$) is $O(n^r)$ if p is a r-variate polynomial of degree n. We will prove this conjecture by induction on the number of variables r. If r is 0 (i.e. p is constant) the conjecture is obviously true. For the induction step let p be a polynomial of the form

$$p = p_n x_1^n + p_{n-1} x_1^{n-1} + \ldots + p_1 x_1 + p_0,$$

where $p_n, \ldots, p_0$ are polynomials containing only $r-1$ variables. The degree of $p_n, \ldots, p_0$ is at most n.

By applying the differentiation rules n times to eliminate x_1, we obtain $n+1$ new $(r-1)$-variate polynomials of a degree smaller or equal than n. Hence, by the induction hypothesis the complexity for proving $p > 0$ is $O(n\, n^{(r-1)}) = O(n^r)$.

References

[Aït85] H. Aït-Kaci. An Algorithm for Finding a Minimal Recursive Path Ordering. *RAIRO*, 19(4):359-382, 1985.

[ACM84] D. S. Arnon, G. E. Collins & S. McCallum. Cylindrical Algebraic Decomposition I. *SIAM Journal of Computing*, 13(4): 865-877, 1984.

[Bel84] F. Bellegarde. Rewriting Systems on FP Expressions that reduce the Number of Sequences they yield. *Symposium on LISP and Functional Programming*, ACM, Austin, TX, 1984.

[BL87] A. Ben Cherifa & P. Lescanne. Termination of Rewriting Systems by Polynomial Interpretations and its Implementation. *Science of Computer Programming*, 9(2):137-159, 1987.

[CL92] A. Cichon & P. Lescanne. Polynomial Interpretations and the Complexity of Algorithms. In *Proc. 11th Int. Conf. on Automated Deduction*, Saratoga Springs, NY, 1992.

[Coh69] P. J. Cohen. Decision Procedures for real and p-adic Fields. *Comm. Pure and Applied Math.*, 22:131-151, 1969.

[Col75] G. E. Collins. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In *Proc. 2nd GI Conf. on Automata Theory and Formal Languages*, Kaiserslautern, Germany, 1975.

[DF85] D. Detlefs & R. Forgaard. A Procedure for automatically proving the Termination of a Set of Rewrite Rules, In *Proc. 1st Int. Conf. on Rewriting Techniques and Applications*, Dijon, France, 1985.

[Der79] N. Dershowitz. A Note on Simplification Orderings. *Information Processing Letters*, 9(5):212-215, 1979.

[Der82] N. Dershowitz. Orderings for Term-Rewriting Systems. *Theoretical Computer Science*, 17:279-301, 1982.

[Der87a] N. Dershowitz. Termination of Rewriting. *Journal of Symbolic Computation*, 3(1, 2):69-115, 1987.

[Der87b] N. Dershowitz. Corrigendum — Termination of Rewriting. *Journal of Symbolic Computation*, 4:409-410, 1987.

[DJ90] N. Dershowitz & J.-P. Jouannaud. Rewrite Systems. *Handbook of Theoretical Comp. Science*, J. van Leuwen, Ed., vol. B, ch. 6, pp. 243-320, Elsevier, 1990.

[DH93] N. Dershowitz & C. Hoot. Topics in Termination. In *Proc. 5th Int. Conf. on Rewriting Techniques and Applications*, Montreal, Canada, 1993.

[Gie95] J. Giesl. Generating Polynomial Orderings for Termination Proofs. In *Proc. 6th Int. Conf. Rewriting Techniques and Applications*, Kaiserslautern, Germany, 1995.

[HL89] D. Hofbauer & C. Lautemann. Termination Proofs and the Length of Derivations (Preliminary Version). In *Proc. 3rd Int. Conf. Rewriting Techniques and Applications*, Chapel Hill, NC, 1989.

[Hon92] H. Hong. Heuristic Search Strategies for Cylindrical Algebraic Decomposition. In *Proc. 1st Int. Conf. on Artificial Intelligence and Symbolic Mathematical Computing*, Karlsruhe, Germany, 1992.

[HL78] G. Huet & D. S. Lankford. On the Uniform Halting Problem for Term Rewriting Systems. Rapport Laboria 283, Institut de Recherche d'Informatique et d'Automatique, Le Chesnay, France, 1978.

[HO80] G. Huet & D. C. Oppen. Equations and Rewrite Rules: A Survey. *Formal Language Theory: Perspectives and Open Problems*, R. Book, ed., Academic Press, pp. 349-405, 1980.

[KB70] D. E. Knuth & P. B. Bendix. Simple Word Problems in Universal Algebras. *Computational Problems in Abstract Algebra*, J. Leech, ed., Pergamon Press, pp. 263-297, 1970.

[Lan75] D. S. Lankford. Canonical Algebraic Simplification in Computational Logic. Memo ATP-25, University of Texas, Austin, TX, 1975.

[Lan76] D. S. Lankford. A Finite Termination Algorithm. Internal Memo, Southwestern University, Georgetown, TX, 1976.

[Lan79] D. S. Lankford. On Proving Term Rewriting Systems are Noetherian. Technical Report Memo MTP-3, Louisiana Tech. Univ., Ruston, LA, 1979.

[Lau88] C. Lautemann. A Note on Polynomial Interpretations. *Bulletin of the European Association for Theoretical Computer Science*, 4:129-131, 1988.

[Les86] P. Lescanne. Divergence of the Knuth-Bendix Completion Procedure and Termination Orderings. *Bulletin of the European Association for Theoretical Computer Science*, 30:80-83, 1986.

[Mar87] U. Martin. How to choose Weights in the Knuth-Bendix Ordering. In *Proc. 2nd Int. Conf. on Rewriting Techniques and Applications*, Bordeaux, France, 1987.

[Mid89] A. Middeldorp. A Sufficient Condition for the Termination of the Direct Sum of Term Rewriting Systems. In *Proc. 4th Annual Symposium on Logic in Computer Science*, Pacific Grove, CA, 1989.

[Pla78] D. A. Plaisted. A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems. Report R-78-943, Univ. Illinois, Urbana, IL, 1978.

[Ste89] J. Steinbach. Extensions and Comparison of Simplification Orderings. *Proc. 3rd Int. Conf. Rewriting Techniques and Applications*, Chapel Hill, NC, 1989.

[Ste91] J. Steinbach. Termination Proofs of Rewriting Systems — Heuristics for Generating Polynomial Orderings. SEKI-Report SR-91-14, Univ. Kaiserslautern, Germany, 1991.

[Ste92] J. Steinbach. Proving Polynomials Positive. In *Proc. 12th Conf. on Foundations of Software Technology and Theoretical Computer Science*, New Delhi, India, 1992.

[Ste93] J. Steinbach. Simplification Orderings: Putting them to the Test. *Journal of Automated Reasoning*, 10(3): 389-397, 1993.

[Ste94] J. Steinbach. *Termination of Rewriting — Extensions, Comparison and Automatic Generation of Simplification Orderings.* Doctoral Dissertation, Univ. Kaiserslautern, Germany, 1994.

[Tar51] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, Berkeley, 1951.