Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT

Proving termination of associative rewriting systems using the Knuth-Bendix ordering

Joachim Steinbach

SEKI Report SR-89-13

Abstract

Term rewriting systems provide a simple mechanism for computing in equations. An equation is converted into a directed rewrite rule by comparing both sides w.r.t. an ordering. However, there exist equations which are incomparable. The handling of such equations includes, for example, partitioning the given equational theory into a set R of rules and a set E of equations. The appropriate reduction relation allows reductions modulo the equations in E. The effective computation with this relation presumes E-termination. Classical termination methods cannot directly guarantee E-termination. This report deals with a new ordering applicable to (R,E)-systems where E contains associative-commutative equations. The method is based on the Knuth-Bendix ordering and is AC-commuting, a property introduced by Jouannaud and Munoz.

Keywords Associative and commutative operators, Associative path ordering, E-commuting, E-compatible, Equational theories, Flattening, Knuth-Bendix ordering, Lexicographical ordering, Multiset ordering, Polynomial ordering, Recursive path ordering, Simplification ordering, Status, Termination, Term rewriting system, Well-founded ordering

Contents

1 Introduction

Term rewriting systems gain more and more in importance because they are a useful model for non-deterministic computations: They are based on directed equations with no explicit control. Various applications in many areas of computer science and mathematics including automatic theorem proving and program verification, abstract data type specifications and algebraic simplification have been developed.
The basic concept of term rewriting systems is that of reducing a given term to an easier one. An equation is converted into a directed rewriting rule in such a way that the right-hand side of the rule is easier than the left-hand side. In order to exclude infinite derivations of terms the rewrite system must terminate. The tools to prove the termination are called orderings. A survey of the most important ones is given in [De87].

The basic idea of an ordering $>$ is to verify that the rewrite relation $\Longrightarrow_{\mathfrak{R}}$ (induced by the rule system $\mathfrak{R}$) is included in $>$. Such an ordering must be well-founded to prevent infinite derivations of terms. To check the inclusion '$\Longrightarrow_{\mathfrak{R}} \subseteq >$' all infinitely many possible derivations must be tested. The key idea is to restrict this infinite test to a finite one by requiring a reduction ordering. A reduction ordering is a well-founded ordering and has the replacement property (also called compatibility with the structure of terms), which means that decreasing a subterm decreases any superterm containing it, too. The notion of reduction orderings leads to the following description of termination of rewrite systems (developed by Lankford, see [De87]):

> A rewrite system $\mathfrak{R}$ terminates if and only if there exists a reduction ordering $>$ such that $\sigma(l) > \sigma(r)$ for each rule $l \longrightarrow_{\mathfrak{R}} r$ and for any substitution σ.

The theorem above reveals another dilemma which is known as the universal quantification on substitutions or the so-called stability w.r.t. substitutions: $s > t$ implies $\sigma(s) > \sigma(t)$, for all σ.

Summarizing, it is to remark that a termination proof of a term rewriting system requires a reduction ordering stabilized w.r.t. substitutions. In general, it is very difficult to guarantee the well-foundedness of a reduction ordering. This fact leads to the basic idea of characterizing classes of orderings for which there is no need to prove this condition. One possible solution is represented by the class of simplification orderings which are at least reduction orderings:

> An ordering is a simplification ordering if and only if it has
> - the replacement property and
> - the subterm property (any term is greater than any of its proper subterms).

Simplification orderings are discussed in detail in [De87]. Well-known simplification orderings are the recursive path orderings and the Knuth-Bendix orderings. Unfortunately, the termination of an arbitrary term rewriting system is an undecidable property, even in the 'one-rule case' ([Da88]).

An additional negative fact derives from the existence of equations of which the left-hand side and the right-hand side are incomparable in any case. For example, a rewriting system containing the commutativity axiom x+y = y+x as a rule is non-terminating. However, if the termination property is not satisfied, the set of axioms can be split into two parts: Those axioms causing non-termination are used as equations E while the others are used as rewrite rules $\mathfrak{R}$. The 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 $t_1 =_E t_1' \Longrightarrow_{\mathfrak{R}} t_2 =_E t_2' \Longrightarrow_{\mathfrak{R}} \cdots$.

We now adapt the general results on termination from the previous page to the case of equational term rewriting systems.

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 to check all derivations of the form $s \Longrightarrow_{\mathfrak{R}/E} t$. This requirement can be refined: If > is E-compatible, then > contains $\Longrightarrow_{\mathfrak{R}/E}$ if and only if it contains $\overset{+}{\Longrightarrow}_{\mathfrak{R}}$ (cf. [BP85]). An ordering > is E-compatible if and only if

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

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

Jouannaud and Munoz succeeded in weakening the E-compatibility for this statement (see [JM84]). They introduced a property called E-commuting:

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

The following theorem (cf. theorem 5.1 on page 11) points out the main importance of E-commutation for the E-termination problem. The theorem is a modification of one contained in [JM84].

> A rewrite system $\mathfrak{R}$ is E-terminating if and only if there exists an E-commuting simplification ordering $>$ such that $\sigma(l) > \sigma(r)$ for each rule $l \longrightarrow_{\mathfrak{R}} r$ and for any substitution σ.

In this report we are going to deal with a special theory E: associative-commutative term rewriting systems. An equational theory E is called an associative-commutative theory if every equation in E is either an associative or commutative axiom:

- $f(x,f(y,z)) = f(f(x,y),z) : f \in \mathfrak{F}_A$ and
- $f(x,y) = f(y,x) : f \in \mathfrak{F}_C$.

In order to describe the fact that f is both associative and commutative we use '$f \in \mathfrak{F}_{AC}$'. An equational term rewriting system $(\mathfrak{R},E)$ will be an associative-commutative term rewriting system if E is an associative-commutative theory.

There only exist a few orderings for this kind of rewriting systems, e.g. the associative path orderings ([Gn88], [GL86], [BP85], [BP85a], [DHJP83]) and the orderings on special polynomial interpretations ([BL87a], [BL86], [La79]).
The polynomial interpretation I for an associative (and commutative) operator must be of the form $I(f)(x,y) = axy + b(x+y) + c$ such that $ac+b-b^2 = 0$. The fundamental disadvantage of polynomial orderings is the difficulty of choosing interpretations for operators such that a given rewrite system terminates.
Associative path orderings 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 distributive axioms. Furthermore, the precedence on the operators has to satisfy a property called associative pair condition. A crucial point of this ordering is its inefficiency which results from the demand that two terms must be pre-processed (flattened and transformed w.r.t. distributive axioms) before they are compared.

Here, we supply a concept which avoids the disadvantages of these two orderings. It is based on the Knuth-Bendix ordering KBO ([KB70]). A modification of this well-known ordering (called associative-commutative Knuth-Bendix ordering, ACK) causes its AC-commutation. The transformation of terms, required by the associative path orderings, is reduced to a minimum. Moreover, the algorithm of [Ma87] for finding an adequate weight function proving the termination of a given rewriting system w.r.t. the KBO can be applied here. The power of the ordering is nearly the same as that of the Knuth-Bendix ordering. Consequently, the applicability of the ACK is bounded by that of the Knuth-Bendix ordering.

After giving some indispensable definitions in the next chapter, the classical Knuth-Bendix ordering of [KB70] will be presented. In chapter 4, we extend this ordering by using the concept of status (see [KL80], [St88]). The definition of the ACK and the proofs of its important properties can be found in chapter 5. Subsequently, we will introduce some extensions of the ACK, e.g. to theories which are only commutative. We conclude with some comments about the comparison of the power of the ACK to other orderings.

2 Notations

A term rewriting system $\mathfrak{R}$ over a set of terms Γ is a finite or countably infinite set of rules, each of the form $l \longrightarrow_{\mathfrak{R}} r$, where l and r are terms in Γ, such that every variable that occurs in r also occurs in l. The set Γ of all terms is constructed from elements of a set $\mathfrak{F}$ of operators (or function symbols) and some denumerably infinite set $\mathfrak{V}$ of variables. The set of ground terms (terms without variables) is denoted by Γ_G. 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 size $|t|$ of a term t is the number of operators and variables occurring in t.

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 set of all positions of a term t is called the set of occurrences and its abbreviation is O(t). We write $t[u \longleftarrow s]$ to denote the term that results from t by replacing t/u (the subterm of t at occurrence u) by s at the occurrence $u \in O(t)$.

A (partial) ordering on Γ_G is a transitive and irreflexive binary relation $>$ and it is called well-founded if there are no infinite descending chains. Most of the orderings on terms are precedence orderings using a special ordering on operators. More precisely, a precedence is a partially ordered set $(\mathfrak{F}, \triangleright)$ consisting of the set $\mathfrak{F}$ of operators and an irreflexive and transitive binary relation $\triangleright$ defined on elements of $\mathfrak{F}$.

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:

$$
\begin{array}{llllll}
(s_1,s_2,\ldots,s_m) & \succ^{lex} & (t_1,t_2,\ldots,t_n) & & & \\
\text{if either} & m > 0 & \wedge & n = 0 & & \\
\quad \text{or} & s_1 \succ t_1 & & & & \\
\quad \text{or} & s_1 = t_1 & \wedge & (s_2,\ldots,s_m) & \succ^{lex} & (t_2,\ldots,t_n).
\end{array}
$$

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 $S \succ\succ T$:

$$
\begin{array}{l}
S \succ\succ T \\
\text{iff} \ \cdot \ S \neq T \ \wedge \\
\qquad \cdot \ (\forall t \in T \setminus S)(\exists s \in S \setminus T) \ s \succ t
\end{array}
$$

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 statuses:

- 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 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 $\{t_1,\ldots,t_n\}$ and otherwise, args(t) delivers the tuple $(t_1,\ldots,t_n)$. Obviously, if the precedence is a quasi-ordering (a transitive and reflexive binary relation), two equivalent symbols w.r.t. the precedence are supposed to have the same status. With this requirement ambiguities will be avoided.

3 Knuth-Bendix ordering

To prove the termination of term rewriting systems we can use the notion of a well-founded set $(S\ ,\ \succ_S)$ which is a set S and a partial ordering $\succ_S$ on S such that any decreasing sequence $e_1 \succ_S e_2 \succ_S \ldots$ of elements of S only consists of a finite number of elements. To construct an ordering we choose a well-founded set $(S\ ,\ \succ_S)$ and a so-called termination function which maps the term algebra into S.

The ordering of Knuth and Bendix (KBO, for short) takes $(\mathbb{N}\ ,\ >)$ as the underlying well-founded set, i.e. it 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. To describe this strategy, we need some pre-requisites and helpful definitions.

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

$$\varphi(c) \geq \varphi_0 \quad \text{if c is a constant,}$$
$$\varphi(f) > 0 \quad \text{if f has one argument.}$$

We extend this weight function on operators to terms. For any term $t = f(t_1,\ldots,t_n)$ let

$$\varphi(t) = \varphi(f) + \sum \varphi(t_i).$$

Definition 3.1 [KB70]

Let $\triangleright$ be a precedence and φ a weight function. The Knuth-Bendix ordering $>_{KBO}$ on terms s and t is defined as follows:

$$
\begin{array}{llllll}
s >_{KBO} t & & & & & \\
\text{iff i)} & (\forall x \in \mathfrak{V})\ \#_x(s) \geq \#_x(t) & \wedge & \varphi(s) > \varphi(t) & & \\
\text{or ii)} & (\forall x \in \mathfrak{V})\ \#_x(s) = \#_x(t) & \wedge & \varphi(s) = \varphi(t) & \wedge & top(s) \triangleright top(t) \\
\text{or iii)} & (\forall x \in \mathfrak{V})\ \#_x(s) = \#_x(t) & \wedge & \varphi(s) = \varphi(t) & \wedge & top(s) = top(t) \\
 & & \wedge & args(s) >^{lex}_{KBO} args(t) & &
\end{array}
$$
∎

The congruence $=_{KBO}$ of this ordering is the syntactical identity.

Note that this ordering is well-founded on ground terms and stable w.r.t. substitutions, i.e. $s > t$ implies $\sigma(s) > \sigma(t)$ for all substitutions σ. The proofs of these properties may be found in [KB70].

Remark 3.2

It is possible to allow one unary operator f with weight zero, *at most*. To guarantee the well-foundedness, all other operators in $\mathfrak{F}$ have to be smaller than f with respect to the precedence (see [KB70]). If the precedence is partial the possibility $f(t) >_{KBO} t$, for all terms t with $f \ntriangleright top(t)$, must be taken into account. ∎

Remark 3.3

Permitting variables, we have to consider each and every one of them as an additional constant symbol uncomparable (w.r.t. $\triangleright$) to all other operators in $\mathfrak{F}$. By admitting a unary operator f with $\varphi(f) = 0$, the possibility that $f^i(x) >_{KBO} x$ must be added to the definition of the KBO. ∎

Example 3.4

Consider the terms

$$s = (x + y) + z \quad \text{and} \quad t = x + (y + z)$$

and the following weight function: $\varphi_0 = 1$ and $\varphi(+) = 0$. We want to prove that $s >_{KBO} t$. Since $\varphi(s) = \varphi(+) + \varphi(x{+}y) + \varphi(z) = 0 + \varphi(+) + \varphi(x) + \varphi(y) + 1 = 0 + 0 + 1 + 1 + 1 = \varphi(t)$ and $top(s) = + = top(t)$ we have to apply the KBO recursively on the tuples of arguments and have to verify $(x{+}y\ ,\ z) >^{lex}_{KBO} (x\ ,\ y{+}z)$. This is true because $x{+}y >_{KBO} x$, since $\varphi(x{+}y) = 2 > 1 = \varphi(x)$ and the variable condition is fulfilled. ∎

The variable condition - $\#_x(s) \geq \#_x(t)$ - guaranteeing the stability w.r.t. substitutions certainly is a very strong restriction. Note that, for example, the distributive law cannot be oriented in the usual direction.

Note that we can use a quasi-ordering on the function symbols instead of a partial ordering. Nevertheless, the KBO remains well-founded.

[Ma87] contains a practical decision procedure for determining whether or not a set of rules can be ordered by a KBO. The basic idea of this algorithm is to transform the desired rules to linear inequalities which are derived from the weight function. The solutions to these inequalities are determined by using the simplex method.

4 An extension of the KBO

The use of the Knuth-Bendix ordering for associative-commutative rewriting systems requires some modifications. This chapter deals with one of these changes: the extension of the KBO. To make the KBO more powerful, we have to realize its short-comings by studying certain examples. In addition, we will analyze the definition of the KBO to point out its weaknesses. The following rules can intuitively be ordered but the KBO does not guarantee their termination.

Example 4.1

We want to prove the termination of the rule

$$s = (y * x) + x \quad \longrightarrow \quad (y + 1) * x = t$$

with the help of the KBO. Therefore, we have to verify that $\varphi(s) \geq \varphi(t)$. This implies that $\varphi(x) = \varphi_o \geq \varphi(1)$. Since the weight of a constant must be greater than or equal to φ_o, the weights of the variables and of 1 have to be the same: $\varphi_o = \varphi(1)$. On this premises, s and t have the same weight. As the multisets of the variables of the two terms are not identical, s can never be greater than t (according to definition 3.1). ∎

One way to establish the termination of this rule is to weaken the variable condition in ii) of definition 3.1 by $(\forall x \in \mathfrak{V})\ \#_x(s) \geq \#_x(t)$ (see [Ma87]).

Example 4.2

We would like to show the termination of the rule

$$s = x * ((-y) * y) \quad \longrightarrow \quad (-(y * y)) * x = t$$

with the help of the KBO. The weights and the leading function symbols of s and t are equal (irrespective of the weight function). This situation demands that $x >_{KBO} -(y * y)$ which cannot be valid (since the variable condition is infringed). ∎

However, $(-y) * y >_{KBO} -(y * y)$ if $* \triangleright -$. In addition to the fact that x is a subterm of s as well as of t, a possible way to guarantee the termination of the rule is to compare the arguments of s and t as multisets instead of lexicographically.

Example 4.3

The termination of the rule

$$(x * y)^2 \longrightarrow x^2 * (-y)^2$$

cannot be proved by any Knuth-Bendix ordering. The reason for this assertion is that two unary function symbols (- and ^{2}) with weight zero must exist since $\varphi(x) + \varphi(*) + \varphi(y) + \varphi(^2) \overset{!}{\geq} \varphi(x) + \varphi(^2) + \varphi(*) + \varphi(-) + \varphi(y) + \varphi(^2)$ which is equivalent to $0 \overset{!}{\geq} \varphi(-) + \varphi(^2)$. This requirement is not allowed (see the remark 3.2 on page 7). ∎

Using a quasi-ordering on the function symbols instead of a partial ordering, we may allow more than one unary operator with weight zero. On the premise that all these operators are equivalent w.r.t. the precedence the induced extension of the KBO also is a well-founded ordering stabilized w.r.t. substitutions (see [St88]). The rule above can then be oriented (if $^2 \triangleright *$). This kind of extension will not be mentioned explicitly by the definition of the new ordering (definition 4.4).

The different aspects of the analysis of the original Knuth-Bendix ordering leads to a new and *more powerful* definition:

Definition 4.4 [St88]

Let $\triangleright$ be a precedence and φ a weight function (as described in chapter 3). The Knuth-Bendix ordering $>_{KBOS}$ with status on terms s and t is defined as

$s >_{KBOS} t$
iff $(\forall x \in \mathfrak{V})\ \#_x(s) \geq \#_x(t) \quad \wedge$
- $\varphi(s) > \varphi(t)$
- $top(s) \triangleright top(t)$
- $args(s) >_{KBOS,\tau(top(s))} args(t)$ ∎

We use a compressed representation of the definition. The hyphens stand for the lexicographical performance of conditions, i.e. $s >_{KBOS} t$ iff $\varphi(s) > \varphi(t)$ or $[\varphi(s) = \varphi(t) \wedge top(s) \triangleright top(t)]$ or $[\varphi(s) = \varphi(t) \wedge top(s) = top(t) \wedge args(s) >_{KBOS,\tau(top(s))} args(t)]$.

Moreover, the index $\tau(f)$ of $>_{KBOS,\tau(f)}$ marks the extension of $>_{KBOS}$ w.r.t. the status of the operator f:

$$\begin{array}{llll} (s_1,\dots,s_m) & >_{KBOS,\tau(f)} & (t_1,\dots,t_n) & \\ \text{iff} & \tau(f) = \text{left} & \wedge & (s_1,\dots,s_m) >^{lex}_{KBOS} (t_1,\dots,t_n) \\ \text{or} & \tau(f) = \text{mult} & \wedge & \{s_1,\dots,s_m\} \gg_{KBOS} \{t_1,\dots,t_n\} \\ \text{or} & \tau(f) = \text{right} & \wedge & (s_m,\dots,s_1) >^{lex}_{KBOS} (t_n,\dots,t_1). \end{array}$$

The KBOS uniquely defines a congruence $\sim$ ($=_{KBOS}$) dependent on $\mathfrak{F}$ and τ via:

$f(s_1,...,s_m) \sim g(t_1,...,t_n)$
iff $f = g \quad \wedge \quad m = n \quad \wedge$
either i) $\tau(f)$ = mult and there is a permutation π of the set $\{1,...,n\}$ such that $s_i \sim t_{\pi(i)}$, for all $i \in [1,n]$,
or ii) $\tau(f) \neq$ mult and $s_i \sim t_i$, for all $i \in [1,n]$.

We conclude this chapter with the enumeration of important properties of the KBOS. The proofs are not given here but may be found in [St88].

Lemma 4.5

KBOS is

- a simplification ordering,
- stable w.r.t. substitutions and
- an extension of the KBO. ∎

5 Handling AC-rewriting

In this chapter we introduce a new class of orderings, associative-commutative Knuth-Bendix orderings, for proving the termination of associative and (or) commutative term rewriting systems. These orderings have to satisfy certain properties. The new Knuth-Bendix ordering will fulfil the conditions required by Jouannaud and Munoz:

Theorem 5.1

Assume that > is a simplification ordering and AC-commuting. Then, $\mathfrak{R}$ is AC-terminating if $\Longrightarrow_{\mathfrak{R}} \subseteq >$.

Proof: This theorem is a slight modification of the following one contained in [JM84]: Assume that > is AC-commuting with $\Longrightarrow_{\mathfrak{R}}$ and contains the homeomorphic embedding relation. Then, $\Longrightarrow_{\mathfrak{R}}$ is AC-terminating.
> is AC-commuting with $\Longrightarrow_{\mathfrak{R}}$ iff for any s', s and t such that $s' =_E s \overset{+}{\Longrightarrow}_{\mathfrak{R}} t$, then $s' > t' =_E t$ for some t'.

We have to prove that the AC-commutation of > together with $\Longrightarrow_{\mathfrak{R}} \subseteq >$ implies the AC-commutation of > with $\Longrightarrow_{\mathfrak{R}}$:

$s' =_E s \overset{+}{\Longrightarrow}_{\mathfrak{R}} t$

⇝ $s' =_E s > t$

because $\Longrightarrow_{\mathfrak{R}} \subseteq >$

⇝ $(\exists t')\ s' > t' =_E t$

since > is AC-commuting

The KBOS is a simplification ordering. Unfortunately, it is not AC-commuting. Consider the following examples.

Example 5.2

$s' = (-(-x)) * x \quad =_{AC} \quad x * (-(-x)) = s$

$>_{KBOS}$

$t' = (-x) * (-x)$

If $* \in \mathfrak{F}_{AC}$, $\tau(*)$ = left and $\varphi(-) > 0$. Assuming the AC-commutation of the KBOS, a term t AC-equivalent to t' must exist which is simpler (w.r.t. the KBOS) than s. The only possible term t is the term t' itself (because no other term exists which is AC-equivalent to t'). But s is not greater than t' (w.r.t. the KBOS). On the contrary, $t' >_{KBOS} s$. ∎

The term s would be greater than t' if the KBOS were used and the status of the multiplication operator were of type multiset. All other relations would be left unchanged.

Taking this idea ($\tau(f)$ = mult if $f \in \mathfrak{F}_C$) into account, let us consider some further examples.

Example 5.3

$$s' = (x \vee y) \vee z \qquad =_{AC} \qquad x \vee (y \vee z) = s$$

$$>_{KBOS}$$

$$t' = \neg(x \mathbin{\overline{\wedge}} y) \vee z$$

If $\vee \in \mathfrak{F}_{AC}$, $\varphi(\vee) = \varphi(\neg) + \varphi(\overline{\wedge})$, $\vee \triangleright \neg$ and $\tau(\vee)$=mult. The term t could be either t' or $z \vee \neg(x \mathbin{\overline{\wedge}} y)$. Unfortunately, $s \not>_{KBOS} t$. The crux of the comparison of s and t is that we have to apply the definition of the KBOS recursively to the arguments of the two terms: There is no subterm of s which is greater than $\neg(x \mathbin{\overline{\wedge}} y)$. ∎

Requiring $\varphi(s') > \varphi(t')$ would avoid the recursive application of the ordering. More precisely, $s' >_{KBOS} t'$ and $s >_{KBOS} t'$ if $\varphi(\vee) > \varphi(\neg) + \varphi(\overline{\wedge})$. Another solution is to forbid that $s' >_{KBOS} t'$ by requiring $\vee$ (all A-operators, respectively) to be minimal w.r.t. the precedence.

Example 5.4

$$s' = ((--x) * x) * y \qquad =_{AC} \qquad (--x) * (x * y) = s$$

$$>_{KBOS}$$

$$t' = ((-x) * (-x)) * y$$

If $* \in \mathfrak{F}_{AC}$, $\tau(*)$=mult and $\varphi(-)>0$. The term t could be either t' or $(-x) * ((-x) * y)$ if we consider the multiset status of $*$. Therefore, we must show that the multiset $\{--x \,,\, x * y\}$ is greater than either $\{(-x) * (-x) \,,\, y\}$ or $\{-x \,,\, (-x) * y\}$. However, no term exists in $\{--x \,,\, x * y\}$ which is either greater than $(-x) * (-x)$ or greater than $(-x) * y$. ∎

A possibility to deal with this problem is to restrict the terms to be compared to their normal forms w.r.t. the associative theory (the flattened versions of the terms).

Usually, terms with AC-operators are represented as flattened terms having no nested occurrences of identical associative operators. This representation requires the operators to have variable arity, i.e. associative function symbols may possess any positive number (> 1) of arguments, whereas non-associative operators have a fixed arity.

Based on this background, the flattening operation fl is defined as follows:

Definition 5.5 [BP85a]

Let be $t = f(t_1,...,t_n)$ a term. Then

$$fl(t) = \begin{cases} t & \text{if } t \text{ is a constant or a variable} \\ f(fl(t_1),...,fl(t_n)) & \text{if } f \notin \mathfrak{F}_A \\ t' & \text{otherwise} \end{cases}$$

with t' results from t by replacing t_i by $fl(t_i)$ if $top(t_i) \neq f$, and replacing t_i by $s_1,...,s_m$ if $fl(t_i) = f(s_1,...,s_m)$. ∎

The following definition of a new Knuth-Bendix ordering summarizes the substantial ideas of the KBO illustrated by the examples.

Definition 5.6

Let φ be a weight function as described in chapter 3 such that $(\forall f \in \mathfrak{F}_A)$ $\varphi(f) = 0$. Furthermore, $\triangleright$ is a precedence such that $(\forall f \in \mathfrak{F}_A) (\nexists g)\ f \triangleright g$. The status function τ fulfils the condition that $\tau(f) = mult$ if $f \in \mathfrak{F}_C$.

The ordering $>_{ACK}$ (associative-commutative Knuth-Bendix ordering) on terms s and t is defined as

$s >_{ACK} t$
iff $fl(s) >_{KBOS} fl(t)$

(with $s =_{ACK} t$ iff $fl(s) =_{KBOS} fl(t)$)

Example 5.7 [BP85], taken from Huet

Consider the following rules:

R1	$x + 0$	$\longrightarrow$	x
R2	$0 + x$	$\longrightarrow$	x
R3	$x * 1$	$\longrightarrow$	x
R4	$1 * x$	$\longrightarrow$	x
R5	$h(0)$	$\longrightarrow$	1
R6	$h(x + y)$	$\longrightarrow$	$h(x) * h(y)$

Assuming + and * are associative and commutative operators, we will prove that the system is terminating with regard to the following preconditions:

symbol	0	1	h	+	*	x,y
φ	2	1	0	0	0	1

$h \triangleright *$ and $\tau(+) = \tau(*) = mult$

The termination of the rules R1 - R4 is proved by the subterm property of the ACK (see 5.10). The rule R5 is terminating since $\varphi(h(0)) = 2 > 1 = \varphi(1)$. The weights of both terms of the rule R6 are identical (= 2) and $h \triangleright *$. ∎

For more examples, see appendix.

The rest of this chapter contains an enumeration of the necessary properties and their proofs which guarantee the use of $>_{ACK}$ for proving the termination of an AC-rewriting system. The following theorem presents a summary of the following lemmata.

Theorem 5.8

ACK is

- a simplification ordering: Lemma 5.9 - 5.12
- stable w.r.t. substitutions: Lemma 5.13 - 5.17
- AC-commuting: Lemma 5.18 - 5.20

Lemma 5.9 ACK is a partial ordering.

Proof: We must show that

i) $t \not>_{ACK} t$ and

ii) $r >_{ACK} s >_{ACK} t$ ~~> $r >_{ACK} t$.

i) Assume that $t >_{ACK} t$

~~> $fl(t) >_{KBOS} fl(t)$

by definition of the ACK

~~> contradiction to the fact that the KBOS is a partial ordering (see [St88])

ii) $r >_{ACK} s >_{ACK} t$

~~> $fl(r) >_{KBOS} fl(s) >_{KBOS} fl(t)$

by definition of the ACK

~~> $fl(r) >_{KBOS} fl(t)$

since the KBOS is a partial ordering (see [St88])

~~> $r >_{ACK} t$

by definition of the ACK ∎

Lemma 5.10 ACK has the subterm property.

Proof: We have to show that $\varepsilon \neq u \in O(t)$ ~~> $t >_{ACK} t/u$. Let us consider a term t and an occurrence $u \in O(t)$, $top(t) = f$ and $top(t/u) = g$. It is obvious that $(\forall x \in \mathfrak{V})$ $\#_x(t) \geq \#_x(t/u)$. We must distinguish two cases which will be proved by induction on $|t|$:

i) $\varphi(fl(t)) > \varphi(fl(t/u))$

~~> $t >_{ACK} t/u$

by definition of the ACK

ii) $\varphi(fl(t)) = \varphi(fl(t/u))$

~~> $\varphi(f) = 0$ $\wedge$ f is a unary operator

- $f \neq g$

~~> $f \triangleright g$

see remark 3.2 on page 7

~~> $t >_{ACK} t/u$

by definition of the ACK

- $f = g$:

  Let $fl(t) = f(t')$, $fl(t/u) = f(t'')$

  $\rightsquigarrow$ $t' >_{KBOS} t''$

  by induction hypothesis since t' (resp. t'') is a proper subterm of $fl(t)$ (resp. $fl(t/u)$)

  $\rightsquigarrow$ $t >_{ACK} t/u$

  by definition of the ACK ∎

Lemma 5.11 ACK has the replacement property.

Proof: It is to show $(\forall i \in [1,n])\ r >_{ACK} s$

$\rightsquigarrow t := f(t_1,\dots,t_i[\varepsilon \leftarrow r],\dots,t_n) >_{ACK} f(t_1,\dots,t_i[\varepsilon \leftarrow s],\dots,t_n) =: t'$.

Obviously, $(\forall x \in \mathfrak{V})\ \#_x(r) \geq \#_x(s) \rightsquigarrow (\forall x \in \mathfrak{V})\ \#_x(t) \geq \#_x(t')$.

We have to examine three cases:

i) $\varphi(fl(r)) > \varphi(fl(s))$

$\rightsquigarrow$ $\varphi(r) > \varphi(s)$

because $\varphi(g) = 0$ if $g \in \mathfrak{F}_A$

$\rightsquigarrow$ $\varphi(t) > \varphi(t')$

$\rightsquigarrow$ $\varphi(fl(t)) > \varphi(fl(t'))$

since $(\forall g \in \mathfrak{F}_A)\ \varphi(g) = 0$

$\rightsquigarrow$ $t >_{ACK} t'$

by definition of the ACK

ii) $\varphi(fl(r)) = \varphi(fl(s)) \wedge top(r) \triangleright top(s)$

$\rightsquigarrow$ $top(r) \notin \mathfrak{F}_A$

since $(\forall g \in \mathfrak{F}_A)\ (\nexists h)\ g \triangleright h$

$\rightsquigarrow$ $fl(t) = f(t_1',\dots,t_m' , fl(r) , \dots)$

α) $top(s) \notin \mathfrak{F}_A \vee top(s) \neq f$

$\rightsquigarrow$ $fl(t') = f(t_1',\dots,t_m' , fl(s) , \dots)$

$\rightsquigarrow$ $\varphi(fl(t)) > \varphi(fl(t'))$

since $\varphi(fl(r)) > \varphi(fl(s))$

$\rightsquigarrow$ $t >_{ACK} t'$

since $fl(r) >_{KBOS} fl(s)$ (regardless of the status of f) and by definition of the ACK

β) $top(s) \in \mathfrak{F}_A \wedge top(s) = f$

$\rightsquigarrow$ $fl(t') = f(t_1',\dots,t_m' , s_1,\dots,s_k , \dots)$

with $fl(s) = f(s_1,\dots,s_k)$

- $\tau(f) = left$

  $\rightsquigarrow$ We must prove that $(t_1',\dots,t_m' , fl(r) , \dots) >_{KBOS,left} (t_1',\dots,t_m' , s_1,\dots,s_k , \dots)$. This is valid because $fl(r) >_{KBOS} fl(s) >_{KBOS} s_1$ and since the KBOS is transitive and has the subterm property.

- $\tau(f)$ = right:
  analogous with the previous case

- $\tau(f)$ = mult
  ~~> We have to show $args(fl(t)) \gg_{KBOS} args(fl(t'))$ which is equivalent to $\{fl(r)\} \gg_{KBOS} \{s_1,\dots,s_k\}$. This is true since $fl(r) >_{KBOS} fl(s)$ and $\{fl(s)\} \gg_{KBOS} \{s_1,\dots,s_k\}$ (because each s_i is a proper subterm of fl(s) and by the transitivity of the KBOS.

iii) $\varphi(fl(r)) = \varphi(fl(s)) \quad \wedge \quad top(r) = top(s) \quad \wedge \quad args(fl(r)) >_{KBOS,\tau(top(r))} args(fl(s))$

α) $top(r) \notin \mathfrak{F}_A \quad \vee \quad top(r) \neq f$
~~> $fl(t) = f(t'_1,\dots,t'_m\ ,\ fl(r)\ ,\ \dots)$ and
$fl(t') = f(t'_1,\dots,t'_m\ ,\ fl(s)\ ,\ \dots)$
~~> $t >_{ACK} t'$
since $fl(r) >_{KBOS} fl(s)$ and regardless of the status of f (see case α of ii)

β) $top(r) \in \mathfrak{F}_A \quad \wedge \quad top(r) = f$
~~> $fl(t) = f(t'_1,\dots,t'_m\ ,\ r_1,\dots,r_k\ ,\ \dots)$ with $fl(r) = f(r_1,\dots,r_k)$ and
$fl(t') = f(t'_1,\dots,t'_m\ ,\ s_1,\dots,s_p\ ,\ \dots)$ with $fl(s) = f(s_1,\dots,s_p)$

- $\tau(f)$ = left
  ~~> We have to show
  $(\exists i \in [1,min(k,p)])\ (\forall j < i)\ \ r_j =_{ACK} s_j \quad \wedge \quad r_i >_{ACK} s_i$.
  This can be proved with the help of the precondition since $args(fl(r)) >_{KBOS,left} args(fl(s))$

- $\tau(f)$ = right:
  analogous with the previous case

- $\tau(f)$ = mult
  ~~> We have to prove that $args(fl(t)) \gg_{KBOS} args(fl(t'))$. This is equivalent to $\{r_1,\dots,r_k\} \gg_{KBOS} \{s_1,\dots,s_p\}$ which is the precondition. ∎

Lemma 5.12 ACK has the deletion property.

Proof: We have to show that $s := f(\dots\ ,\ r\ ,\ \dots) >_{ACK} f(\dots\ ,\ \dots) =: t$. It is obvious that $(\forall x \in \mathfrak{V})\ \#_x(s) \geq \#_x(t)$.
~~> $(\forall x \in \mathfrak{V})\ \#_x(fl(s)) \geq \#_x(fl(t))$ (*)
since the fl-operator does not remove leaves of the terms

It is also obvious that $\varphi(s) > \varphi(t)$ because t results from s by removing at least one leaf (and the constant symbols and variables have a positive weight).

⇝ $\varphi(fl(s)) > \varphi(fl(t))$
since $(\forall t)\ \varphi(t) = \varphi(fl(t))$ (because $\varphi(g) = 0$ if $g \in \mathfrak{F}_A$)

⇝ $fl(s) >_{KBOS} fl(t)$
since (*) is valid

⇝ $s >_{ACK} t$
by definition of the ACK ∎

Lemma 5.13 $(\forall t \in \Gamma)\ (\forall \sigma)\ \ fl(\sigma(fl(t))) = fl(\sigma(t))$

Proof: $fl(fl(t)) = fl(t)$
⇝ $t =_A fl(t)$
since $s =_A t$ iff $fl(s) = fl(t)$
⇝ $(\forall \sigma)\ \sigma(t) =_A \sigma(fl(t))$
since $=_A$ is closed under substitution
⇝ $fl(\sigma(t)) = fl(\sigma(fl(t)))$
since $s =_A t$ iff $fl(s) = fl(t)$ ∎

Lemma 5.14 $(\forall \sigma)\ \ fl(s) =_{KBOS} fl(t)$ ⇝ $fl(\sigma(s)) =_{KBOS} fl(\sigma(t))$

Proof: $fl(s) =_{KBOS} fl(t)$
⇝ $fl(s) \sim fl(t)$
by definition of the KBOS
⇝ $s =_{AC} t$
see [GD88], proposition 5
⇝ $\sigma(s) =_{AC} \sigma(t)$
⇝ $fl(\sigma(s)) \sim fl(\sigma(t))$
see [GD88], proposition 5
⇝ $fl(\sigma(s)) =_{KBOS} fl(\sigma(t))$
by definition of the KBOS ∎

Lemma 5.15 Let be t a term, $fl(t) = f(t_1',\ldots,t_k')$ its flattened version and $\sigma = \{x \leftarrow g(s_1,\ldots,s_m)\}$ a substitution. Then, $fl(\sigma(t)) = f(fl(\sigma(t_1')),\ldots,fl(\sigma(t_k')))$ if $f \neq g$ or $(\forall i \in [1,k])\ t_i' \neq x$.

Proof:

$fl(\sigma(t))$	$= fl(\sigma(fl(t)))$	by using lemma 5.13
	$= fl(\sigma(f(t_1',\ldots,t_k')))$	by precondition
	$= fl(f(\sigma(t_1'),\ldots,\sigma(t_k')))$	by definition of substitutions

i) $f \neq g$:

a) $t_i' = x$
$\leadsto$ $\sigma(t_i') = g(s_1,\dots,s_m)$
$\leadsto$ assertion
because $f \neq g$

b) $t_i' \neq x$
$\leadsto$ $top(\sigma(t_i')) \neq f$
since $fl(t)$ is flattened
$\leadsto$ assertion

ii) $(\forall i \in [1,k])\ t_i' \neq x$:
analogous with b) of i) ∎

Lemma 5.16 $fl(s) >_{KBOS} fl(t) \leadsto fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
if $\sigma = \{x \leftarrow f(r_1, r_2)\}$, $f \in \mathfrak{F}_A$

Proof: It is obvious that $(\forall x \in \mathfrak{V})\ \#_x(fl(\sigma(s))) \geq \#_x(fl(\sigma(t)))$ if $(\forall x \in \mathfrak{V})\ \#_x(fl(s)) \geq \#_x(fl(t))$ since lemma 6.14 of [St88] is valid.
We have to consider three cases which will be proved by induction on $|fl(s)| + |fl(t)|$.

i) $\varphi(fl(s)) > \varphi(fl(t))$
$\leadsto$ $\varphi(fl(\sigma(s))) > \varphi(fl(\sigma(t)))$
since $(\forall t \in \Gamma)\ \varphi(fl(\sigma(t))) = \varphi(\sigma(fl(t)))$ (because $\varphi(f) = 0$, $f \in \mathfrak{F}_A$)
and lemma 6.14 of [St88]

$\leadsto$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
by definition of the KBOS

ii) $\varphi(fl(s)) = \varphi(fl(t)) \ \wedge\ top(fl(s)) \triangleright top(fl(t))$
$\leadsto$ $\varphi(fl(\sigma(s))) \geq \varphi(fl(\sigma(t)))$
since $(\forall t \in \Gamma)\ \varphi(fl(\sigma(t))) = \varphi(\sigma(fl(t)))$ (because $\varphi(f) = 0$, $f \in \mathfrak{F}_A$)
and lemma 6.14 of [St88]

$\leadsto$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
because $top(fl(\sigma(s))) \triangleright top(fl(\sigma(t)))$ and by definition of the KBOS

iii) $\varphi(fl(s)) = \varphi(fl(t)) \ \wedge\ top(fl(s)) = top(fl(t))$
$\leadsto$ $args(fl(s)) >_{KBOS,\tau(top(s))} args(fl(t))$
since $fl(s) >_{KBOS} fl(t)$ and by definition of the KBOS

$\leadsto$ $\varphi(fl(\sigma(s))) \geq \varphi(fl(\sigma(t))) \ \wedge\ top(fl(\sigma(s))) = top(fl(\sigma(t)))$
since $(\forall t \in \Gamma)\ \varphi(fl(\sigma(t))) = \varphi(\sigma(fl(t)))$ (because $\varphi(f) = 0$, $f \in \mathfrak{F}_A$)
and lemma 6.14 of [St88]

α) $\varphi(fl(\sigma(s))) > \varphi(fl(\sigma(t)))$

$\leadsto$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
by definition of the KBOS

β) $\varphi(fl(\sigma(s))) = \varphi(fl(\sigma(t))) \quad \wedge \quad \tau(top(fl(s))) = left$:

Let be $s = g(s_1,...,s_k)$ and $t = g(t_1,...,t_k)$,
$fl(s) = g(s_1',...,s_m')$ and $fl(t) = g(t_1',...,t_n')$

$\leadsto$ $(\exists i)\ (\forall j < i)\ \ s_j' =_{KBOS} t_j' \quad \wedge \quad s_i' >_{KBOS} t_i'$ (*)

I) $g \neq f \quad \vee \quad [(\nexists p \in [1,max(m,n)])\ s_p' = x \quad \vee \quad t_p' = x]$

$\leadsto$ $fl(\sigma(s)) = g(fl(\sigma(s_1')),...,fl(\sigma(s_m')))$ and
$fl(\sigma(t)) = g(fl(\sigma(t_1')),...,fl(\sigma(t_n')))$
by using lemma 5.15

$\leadsto$ $(\forall j < i)\ fl(\sigma(s_j')) =_{KBOS} fl(\sigma(t_j')) \quad \wedge$
$fl(\sigma(s_i')) >_{KBOS} fl(\sigma(t_i'))$
since
- $s_p' = fl(s_p') \quad \wedge \quad t_p' = fl(t_p')$, $\forall p \in [1,k]$ because $g(s_1',...,s_m')$ and $g(t_1',...,t_n')$ are the flattened versions of s and t , respectively.
- $fl(s) =_{KBOS} fl(t) \leadsto fl(\sigma(s)) =_{KBOS} fl(\sigma(t))$, $\forall\sigma$ (with the help of lemma 5.14)
- lemma 5.13

and by using the induction hypothesis together with (*)

$\leadsto$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
by definition of the KBOS

II) $g = f \quad \wedge \quad [(\exists p \in [1,max(m,n)])\ \ s_p' = x \quad \vee \quad t_p' = x]$

$\leadsto$ $fl(s) = f(s_1',...,s_m')$ and $fl(t) = f(t_1',...,t_n')$

Note that (*) is valid. Let be $fl(\sigma(s)) = f(s_1'',...,s_p'')$, $fl(\sigma(t)) = f(t_1'',...,t_q'')$.

- $s_i' \neq x \quad \wedge \quad t_i' \neq x$

  $\leadsto$ $(\exists a \in [1,min(p,q)])\ \ (\forall j < a)\ \ \ s_j'' =_{KBOS} t_j'' \quad \wedge$
  $s_a'' >_{KBOS} t_a''$
  since $s_a'' = fl(\sigma(s_i'))$, $t_a'' = fl(\sigma(t_i'))$, lemma 5.13 and by induction hypothesis

  $\leadsto$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
  by definition of the KBOS

- $t_i' = x$

  $\leadsto$ $(\exists u \in O(s_i'))\ \ \ u \neq \varepsilon \quad \wedge \quad s_i'/u = x$
  since $s_i' >_{KBOS} t_i'$ and by definition of the KBOS

  $\leadsto$ $\varphi(s_i') > \varphi(t_i') \quad \vee \quad [s_i' = h(h(...h(x)...) \quad \wedge \quad \varphi(h) = 0]$
  by definition of the KBOS

$\rightsquigarrow$ $(\exists a \in [1,\min(p,q)])$ $(\forall j < a)$ $s_j'' =_{KBOS} t_j''$ $\wedge$ $s_a'' >_{KBOS} t_a''$
since $s_a'' = fl(\sigma(s_i'))$, $fl(\sigma(x)) = f(t_a'',...)$ and lemma 6.14 of [St88] and either $\varphi(s_a'') > \varphi(t_a'')$ or with the help of remark 3.2 / 3.3

$\rightsquigarrow$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
by definition of the KBOS

γ) $\varphi(fl(\sigma(s))) = \varphi(fl(\sigma(t)))$ $\wedge$ $\tau(top(fl(s))) = right$:
analogous with β)

δ) $\varphi(fl(\sigma(s))) = \varphi(fl(\sigma(t)))$ $\wedge$ $\tau(top(fl(s))) = mult$
$\rightsquigarrow$ $args(fl(s)) \gg_{KBOS} args(fl(t))$ (∗∗)
since $fl(s) >_{KBOS} fl(t)$ and by definition of the KBOS

I) $g \neq f$ $\vee$ $[(\nexists p \in [1,k])$ $s_p' = x$ $\vee$ $t_p' = x]$
$\rightsquigarrow$ $fl(\sigma(s)) = g(fl(\sigma(s_1')),...,fl(\sigma(s_m')))$ and
$fl(\sigma(t)) = g(fl(\sigma(t_1')),...,fl(\sigma(t_n')))$
by using lemma 5.15

$\rightsquigarrow$ $args(fl(\sigma(s))) \gg_{KBOS} args(fl(\sigma(t)))$
since
- $s_p' = fl(s_p')$ $\wedge$ $t_p' = fl(t_p')$, $\forall p$ because $g(s_1',...,s_m')$ and $g(t_1',...,t_n')$ are the flattened versions of s and t, respectively.
- $fl(s) =_{KBOS} fl(t)$ $\rightsquigarrow$ $fl(\sigma(s)) =_{KBOS} fl(\sigma(t))$ with the help of lemma 5.14

and by using the induction hypothesis together with (∗∗)

$\rightsquigarrow$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
by definition of the KBOS

II) $g = f$ $\wedge$ $[(\exists p \in [1,\max(m,n)])$ $s_p' = x$ $\vee$ $t_p' = x]$
$\rightsquigarrow$ $fl(s) = f(s_1',...,s_m')$ and $fl(t) = f(t_1',...,t_n')$

$\rightsquigarrow$ $S := \{s_1',...,s_m'\} \gg_{KBOS} \{t_1',...,t_n'\} =: T$
by definition of the KBOS (see (∗∗))

$\rightsquigarrow$ $(\forall t_j' \in T\backslash S)$ $(\exists s_i' \in S\backslash T)$ $s_i' >_{KBOS} t_j'$

- $s_i' \neq x$ $\wedge$ $t_j' \neq x$
$\rightsquigarrow$ $top(fl(\sigma(s_i'))) \neq f$ $\wedge$ $top(fl(\sigma(t_j'))) \neq f$
since $fl(s_i') = s_i'$ $\wedge$ $fl(t_j') = t_j'$

$\leadsto$ $fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
since $fl(\sigma(s_i')) >_{KBOS} fl(\sigma(t_j'))$ (by induction hypothesis) and with the help of the lemmata 5.13 and 5.14

- $t_j' = x$

$\leadsto$ $(\exists u \in O(s_i'))\ u \neq \varepsilon \quad \wedge \quad s_i'/u = x$
since $s_i' >_{KBOS} t_j'$ and by definition of the KBOS

$\leadsto$ $\varphi(s_i') > \varphi(t_j') \quad \vee \quad [s_i' = h(h(...h(x)...) \quad \wedge \quad \varphi(h) = 0]$
by definition of the KBOS

Let be $fl(\sigma(x)) = f(r_1',...,r_p')$. It is sufficient to show that $fl(\sigma(s_i')) >_{KBOS} r_q'$, for all $q \in [1,p]$:
This is true since the variable condition is fulfilled and $\varphi(fl(\sigma(s_i'))) > \varphi(r_q')$
because r_q' is a subterm of $\varphi(s_i'))$ and $\varphi(s_i') > \varphi(t_j')$ ∎

Lemma 5.17 ACK is stable w.r.t. substitutions.

Proof: We have to show that $(\forall\sigma)\ s >_{ACK} t \leadsto \sigma(s) >_{ACK} \sigma(t)$.

$s >_{ACK} t$

$\leadsto$ $fl(s) >_{KBOS} fl(t)$
by definition of the ACK

$\leadsto$ $(\forall\sigma)\ fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
with the help of the following facts:

- $fl(s) >_{KBOS} fl(t) \leadsto fl(\sigma(s)) >_{KBOS} fl(\sigma(t))$
  if $\sigma = \{x \leftarrow f(r_1,...,r_n)\}$, $f \notin \mathfrak{F}_A$
  This is true because $(\forall t \in \Gamma)\ fl(\sigma(t)) = \sigma(fl(t))$ and by using the stability of the KBOS w.r.t. substitutions (see [St88]).

- lemma 5.16

- Let be σ a substitution whose domain is $\{x_1,...,x_n\}$. Then $\sigma = \sigma_1 \dots \sigma_n$ where σ_i is the elementary substitution (the domain is reduced to a single variable) whose domain is $\{x_i\}$. The proof can be found in [GL86].

$\leadsto$ $(\forall\sigma)\ \sigma(s) >_{ACK} \sigma(t)$
by definition of the ACK ∎

Lemma 5.18 $s =_{AC} t \rightsquigarrow s =_{ACK} t$

Proof: $s =_{AC} t$

$\Leftrightarrow$ $fl(s) \sim fl(t)$
see [GD88], proposition 5

$\Leftrightarrow$ $fl(s) =_{KBOS} fl(t)$
by definition of the KBOS

$\Leftrightarrow$ $s =_{ACK} t$
by definition of the ACK ∎

Lemma 5.19 $r =_{KBOS} s >_{KBOS} t \rightsquigarrow r >_{KBOS} t$

Proof: We will show it by induction on $|t|$. It is clear that $(\forall x \in \mathfrak{V})\ \#_x(r) \geq \#_x(t)$ if $(\forall x \in \mathfrak{V})\ \#_x(r) = \#_x(s) \geq \#_x(t)$ because $>$ on $\mathbb{N}$ is a partial ordering. We have to consider five disjoint cases:

i) $\varphi(s) > \varphi(t)$

$\rightsquigarrow$ $\varphi(r) > \varphi(t)$
since $\varphi(r) = \varphi(s)$ which follows from $r =_{KBOS} s$

$\rightsquigarrow$ $r >_{KBOS} t$
by definition of the KBOS

ii) $\varphi(s) = \varphi(t) \wedge top(s) \triangleright top(t)$

$\rightsquigarrow$ $\varphi(r) = \varphi(t) \wedge top(r) \triangleright top(t)$
since $\varphi(r) = \varphi(s)$ ($\Leftarrow r =_{KBOS} s$)
and $top(r) = top(s) \wedge \triangleright$ is a partial ordering

$\rightsquigarrow$ $r >_{KBOS} t$
by definition of the KBOS

iii) $\varphi(s) = \varphi(t) \wedge top(s) = top(t) \wedge \tau(top(s)) = left$:

Let be $s = f(s_1,\dots,s_m)$ and $t = f(t_1,\dots,t_n)$ ($\rightsquigarrow r = f(r_1,\dots,r_m)$)

$\rightsquigarrow$ $(s_1,\dots,s_m) >_{KBOS,left} (t_1,\dots,t_n)$
since $s >_{KBOS} t$ and by definition of the KBOS

$\rightsquigarrow$ $(\exists i)\ (\forall j < i)\ s_j =_{KBOS} t_j \wedge s_i >_{KBOS} t_i \vee [(\forall i \in [1,n])\ s_i =_{KBOS} t_i \wedge m > n]$
by definition of $>_{KBOS,left}$

$\rightsquigarrow$ $(\forall j < i)\ r_j =_{KBOS} t_j \wedge r_i >_{KBOS} t_i$
because
- $(\forall j \leq i)\ r_j =_{KBOS} s_j$
  since $r =_{KBOS} s$ and by definition of the KBOS
- $t_1 =_{KBOS} t_2 =_{KBOS} t_3 \rightsquigarrow t_1 =_{KBOS} t_3$
  can be proved easily

- induction hypothesis
  since $r_i =_{KBOS} s_i >_{KBOS} t_i$

$\leadsto$ $r >_{KBOS} t$
by definition of the KBOS

iv) $\varphi(s) = \varphi(t) \quad \wedge \quad top(s) = top(t) \quad \wedge \quad \tau(top(s)) = right$:
analogous with the previous case

v) $\varphi(s) = \varphi(t) \quad \wedge \quad top(s) = top(t) \quad \wedge \quad \tau(top(s)) = mult$:
Let be $s = f(s_1,\dots,s_m)$ and $t = f(t_1,\dots,t_n)$
$\leadsto$ $\{s_1,\dots,s_m\} \gg_{KBOS} \{t_1,\dots,t_n\}$
since $s >_{KBOS} t$ and by definition of the KBOS
w.l.o.g. let be $\{s_1,\dots,s_m\} \cap \{t_1,\dots,t_n\} = \emptyset$

$\leadsto$ $(\forall t_j)\ (\exists s_i)\ s_i >_{KBOS} t_j$
by definition of the extension of the KBOS to multisets
$\leadsto$ It is sufficient to show that
$\alpha)$ $s_i >_{KBOS} t_j \leadsto (\exists k)\ r_k >_{KBOS} t_j$ and
$\beta)$ $(\nexists k)\ r_i =_{KBOS} t_k$

$\alpha)$ $(\exists k)\ r_k =_{KBOS} s_i$
since $r =_{KBOS} s$ and by definition of the KBOS
$\leadsto$ $r_k >_{KBOS} t_j$
because $s_i >_{KBOS} t_j$ and with the help of the induction hypothesis

$\beta)$ Assume that $(\exists k)\ r_i =_{KBOS} t_k$
$\leadsto$ $t_k =_{KBOS} s_i$
since $r_i =_{KBOS} s_i$ $\quad (\Leftarrow r =_{KBOS} s)$
$\leadsto$ $\{s_1,\dots,s_m\} \cap \{t_1,\dots,t_n\} \neq \emptyset$
which is a contradiction to the precondition ∎

Lemma 5.20 ACK is AC-commuting.

Proof: We have to show that $r =_{AC} s >_{ACK} t \leadsto (\exists t')\ r >_{ACK} t' =_{AC} t$.
$r =_{AC} s >_{ACK} t$
$\leadsto$ $r =_{ACK} s >_{ACK} t$
by using lemma 5.18
$\leadsto$ $fl(r) =_{KBOS} fl(s) >_{KBOS} fl(t)$
by definition of the ACK

$\leadsto$ $fl(r) >_{KBOS} fl(t)$
with the help of lemma 5.19

$\leadsto$ $r >_{ACK} t$
by definition of the ACK

$\leadsto$ $t' = t$
since $t' = t$ $\leadsto$ $t' =_{AC} t$ ∎

6 Improving the ACK

The associative-commutative Knuth-Bendix ordering defined in the last chapter is an ordering which can prove the termination of rewriting systems modulo an associative and commutative theory. This theoretical aspect is the foundation for using the ACK in practice. However, from a practical point of view the ACK is inefficient since the terms to be compared must be flattened. Subsequently, we will present two different kinds of versions of the ACK which improve its applicability.

6.1 Reducing the use of the fl-operator

Comparing two terms w.r.t. the ACK we do not always have to flatten them. This expense will only be necessary if the *arguments* of both terms must be compared. The following **lemma** reifies this fact:

> $s >_{ACK} t$
> iff $(\forall x \in \mathfrak{V})\ \#_x(s) \geq \#_x(t) \quad \wedge$
> - $\varphi(s) > \varphi(t)$
> - $top(s) \triangleright top(t)$
> - $args(fl(s)) >_{ACK,\tau(top(s))} args(fl(t))$
>
> with all conditions of 5.6 about φ and $\triangleright$ (see page 13)

Proof: The proof can be easily performed by using the following facts:

- $\#_x(fl(s)) \geq \#_x(fl(t))$ iff $\#_x(s) \geq \#_x(t)$
  by definition of the fl-operator
- $\varphi(fl(s)) > \varphi(fl(t))$ iff $\varphi(s) > \varphi(t)$ and $\varphi(fl(s)) = \varphi(fl(t))$ iff $\varphi(s) = \varphi(t)$
  since $\varphi(f) = 0$ if $f \in \mathfrak{F}_A$
- $top(fl(t)) = top(t),\ \forall t \in \Gamma$ and $\triangleright$ is a partial ordering ∎

Note that it is even possible to improve this version by only

- flattening the term if $top(s) \in \mathfrak{F}_A$ and
- flattening the highest level of s and t if $top(s) \in \mathfrak{F}_A$.

6.2 Restricting the ACK to C-theories

Up to now we admitted commutative *and* associative operators. The exclusion of the latter enables us to simplify the ACK: The flattening of the terms to be compared is completely redundant. Obviously, we do not need to check the conditions about associative operators.

Definition 6.2.1

Let be $\triangleright$ a precedence and φ a weight function as described in chapter 3. The status function τ fulfils the condition that $\tau(f) = \text{mult}$ if $f \in \mathfrak{F}_C$.
The ordering $>_{CK}$ (commutative Knuth-Bendix ordering) on terms s and t is defined as

$s >_{CK} t$
iff $s >_{KBOS} t$ ∎

We want to use this restricted version of the KBOS as an ordering to prove the termination of rewrite systems modulo commutative theories. Therefore, the CK must be a simplification ordering and C-commuting. The proofs of these properties will follow.

Lemma 6.2.2

CK is a simplification ordering and stable w.r.t. substitutions.

Proof: This is valid since the KBOS has the same properties (lemma 4.5 on page 10). ∎

Lemma 6.2.3

CK is C-commuting.

Proof: We have to show that $s' =_C s >_{CK} t \leadsto (\exists t')\ s' >_{CK} t' =_C t$.

$s' =_C s >_{CK} t$
$\leadsto$ $s' =_{CK} s >_{CK} t$
since $s =_C t \leadsto s =_{CK} t$ (because $\tau(f) = \text{mult}$ if $f \in \mathfrak{F}_C$)
$\leadsto$ $s' =_{KBOS} s >_{KBOS} t$
by definition of the CK
$\leadsto$ $s' >_{KBOS} t$
lemma 5.19
$\leadsto$ $s' >_{CK} t$
by definition of the CK
$\leadsto$ $t' = t$
since $t' = t \leadsto t' =_C t$ ∎

7 Conclusion

This paper introduces a class of termination orderings for associative and (or) commutative term rewriting systems, called associative-commutative Knuth-Bendix orderings (ACK, for short). The ACK is a modified version of the Knuth-Bendix ordering with the following basic concepts:

- Extending the KBO to KBOS by permitting various statuses (to compare the arguments of two terms),
- Assigning multiset status to each commutative function symbol,
- Assigning weight zero to each associative operator which has to be minimal w.r.t. the precedence and
- *Partly* flattening the terms to be compared.

This ordering can prove the AC-termination of a set of rules since it is a simplification ordering and AC-commuting (see [JM84]). A great deal of the substantial aspects of this ordering are similar to those of the associative path ordering.

The power of the ACK is approximately the same as that of the KBOS. We expect this conjecture to be confirmed by several tests.

Unlike the associative path ordering APO (cf. [Gn88], [GL86], [BP85], [BP85a], [DHJP83]), the ACK does not require a complete transformation (including distributing and flattening) of the terms to compare. The comparison of the power of the APO and the ACK leads to the fact that they are incomparable:

ACK is more powerful than the APO:	APO is more powerful than the ACK:
$x^z * y^z \quad >_{ACK} \quad (x * y)^z$	$s(x) * y \quad >_{APO} \quad y + (x * y)$
but	but
$x^z * y^z \quad \not>_{APO} \quad (x * y)^z$	$s(x) * y \quad \not>_{ACK} \quad y + (x * y)$

with $*, + \in \mathfrak{F}_{AC}$, $\varphi(exp) > 0$, $* \triangleright +$

There exists another ordering for AC-termination: a restricted version of the ordering on polynomial interpretations (POL, for short). The power of this method and the power of the ACK also overlap:

ACK is more powerful than the POL:	POL is more powerful than the ACK:
$(-x) + x \quad >_{ACK} \quad x + (-x)$ but $(-x) + x \quad \not>_{POL} \quad x + (-x)$	$s(x) * y \quad >_{POL} \quad y + (y * x)$ but $s(x) * y \quad \not>_{ACK} \quad y + (y * x)$

with $+ \in \mathfrak{F}_A$, $\tau(+) = \text{left}$, $I(+)(x,y) = x+y$, $I(*)(x,y) = x+3y$, $I(s)(x) = 4x$

From a practical point of view, the POL is applicable to more rules than the ACK, but it is very difficult to choose the adequate interpretations for the operators (see [BL87a]). On the contrary, it is easy to determine whether or not a set of rules can be ordered by a Knuth-Bendix ordering (cf. [Ma87]).

The generalization of the presented method for other theories as well as the weakening of the conditions for associative operators (by eventually distributing simultaneously) will be part of future plans.

Acknowledgement

There remains the pleasant duty to thank those who somehow co-operated in forming this paper: Jürgen Avenhaus, Jörg Denzinger, Roland Fettig, Bernhard Gramlich, Rita Kohl, Klaus Madlener, Inger Sonntag and Michael Zehnter.

References

[Ad87] Christian Adler
Vervollständigung von Termersetzungssystemen modulo einer gleichungsdefinierten Theorie
Master Thesis, Kaiserslautern, W. Germany, 1987

[BD87] Leo Bachmair / Nachum Dershowitz
Completion for rewriting modulo a congruence
Proc. 2nd RTA, Bordeaux, France, 1987, LNCS 256

[BD86] Leo Bachmair / Nachum Dershowitz
Commutation, transformation and termination
Proc. 8th CADE, Oxford, U.K., 1986, LNCS 230

[BL87] Francoise Bellegarde / Pierre Lescanne
Transformation orderings
Proc. TAPSOFT, Pisa, Italy, 1987, LNCS 249

[BL87a] Ahlem Ben Cherifa / Pierre Lescanne
Termination of rewriting systems by polynomial interpretations and its implementation
Science of Computer Programming 9 (2), 1987

[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
Proc. 8th CADE, Oxford, U.K., 1986, LNCS 230

[BP85] Leo Bachmair / David A. Plaisted
Termination orderings for associative-commutative rewriting systems
J. Symbolic Computation 1, 1985

[BP85a] Leo Bachmair / David A. Plaisted
Associative path orderings
Proc. 1st RTA, Dijon, France, 1985, LNCS 202

[Da88] Max Dauchet
Termination of rewriting is undecidable in the one-rule case
MFCS, Carlsbad, CSSR, 1988, LNCS 324

[De87] Nachum Dershowitz
Termination of rewriting
J. Symbolic Computation 3, 1987

[DHJP83] Nachum Dershowitz / Jieh Hsiang / N. Alan Josephson / David A. Plaisted
Associative-commutative rewriting
Proc. 8th IJCAI, Karlsruhe, W. Germany, 1983

[GD88] Bernhard Gramlich / Jörg Denzinger
Efficient AC-matching using constraint propagation
SEKI-Report, Kaiserslautern, W. Germany, 1988

[Gn88] Isabelle Gnaedig
Total orderings for equational theories
Working document, Nancy, France, 1988

[GL86] Isabelle Gnaedig / Pierre Lescanne
Proving termination of associative-commutative rewriting systems by rewriting
Proc. 8th CADE, Oxford, U.K., 1986, LNCS 230

[Jo83] Jean-Pierre Jouannaud
Confluent and coherent equational term rewriting systems - application to proofs in abstract data types
Proc. CAAP, L'Aquila, Italy, 1983, LNCS 159

[JK86] Jean-Pierre Jouannaud / Helene Kirchner
Completion of a set of rules modulo a set of equations
SIAM J. Computing 15 (4), 1986

[JKR83] Jean-Pierre Jouannaud / Helene Kirchner / Jean-Luc Remy
Church-Rosser properties of weakly terminating equational term rewriting systems
Proc. 8th IJCAI, Karlsruhe, W. Germany, 1983

[JM84] Jean-Pierre Jouannaud / Miguel Munoz
Termination of a set of rules modulo a set of equations
Proc. 7th CADE, Napa, California, 1984, LNCS 170

[KB70] Donald E. Knuth / Peter B. Bendix
Simple word problems in universal algebras
Computational Problems in abstract algebra, Pergamon Press, 1970

[KL80] Sam Kamin / Jean-Jacques Levy
Attempts for generalizing the recursive path orderings
Unpublished manuscript, Urbana, Illinois, 1980

[La79] Dallas S. Lankford
On proving term rewriting systems are noetherian
Memo MTP-3, Ruston, Louisiana, 1979

[Ma87] Ursula Martin
How to choose the weights in the Knuth-Bendix ordering
Proc. 2nd RTA, Bordeaux, France, 1987, LNCS 256

[Pl83] David A. Plaisted
An associative path ordering
Proc. NFS Workshop on the RRL, Schenectady, U.S.A., 1983

[PS81] Gerald E. Peterson / Mark E. Stickel
Complete sets of reductions for some equational theories
J. ACM 28 (2), 1981

[St88] Joachim Steinbach
Term orderings with status
SEKI-Report, Kaiserslautern, W. Germany, 1988
also: **Extensions and Comparison of Simplification orderings**
Proc. 3rd RTA, Chapel Hill, North Carolina, 1989, LNCS 355

[Ze89] Michael Zehnter
Theorieverträgliche Ordnungen - Eine Übersicht
Project report, Kaiserslautern, W. Germany, 1989

Appendix: Examples

Associativity and Endomorphism

$\mathfrak{R}$:	$f(x) + f(y)$	$\rightarrow$	$f(x + y)$	
	$f(x) + (f(y) + z)$	$\rightarrow$	$f(x + y) + z$	$\varphi(f) > 0$
E:	$(x + y) + z$	$=$	$x + (y + z)$	
	$x + y$	$=$	$y + x$	

Abelian group theory

$\mathfrak{R}$:	$x + 0$	$\rightarrow$	x	
	$x + i(x)$	$\rightarrow$	0	
	$i(0)$	$\rightarrow$	0	$\varphi(i) = 0$
	$i(i(x))$	$\rightarrow$	x	$i \triangleright +$
	$i(x + y)$	$\rightarrow$	$i(x) + i(y)$	$i \triangleright 0$
E:	$(x + y) + z$	$=$	$x + (y + z)$	
	$x + y$	$=$	$y + x$	

Disjunctive normal form (Dershowitz)

$\mathfrak{R}$:	$\neg\neg x$	$\rightarrow$	x	
	$\neg(x \vee y)$	$\rightarrow$	$\neg\neg\neg x \wedge \neg\neg\neg y$	
	$\neg(x \wedge y)$	$\rightarrow$	$\neg\neg\neg x \vee \neg\neg\neg y$	
	$x \wedge x$	$\rightarrow$	x	
	$x \vee x$	$\rightarrow$	x	$\varphi(\neg) = 0$
				$\neg \triangleright \wedge$
E:	$(x \wedge y) \wedge z$	$=$	$x \wedge (y \wedge z)$	$\neg \triangleright \vee$
	$x \wedge y$	$=$	$y \wedge x$	
	$(x \vee y) \vee z$	$=$	$x \vee (y \vee z)$	
	$x \vee y$	$=$	$y \vee x$	

Unary integer addition (Dershowitz)

$\mathfrak{R}$:	$x + 0$	$\rightarrow$	x
	$0 + y$	$\rightarrow$	y
	-0	$\rightarrow$	0
	$-((-x) + y)$	$\rightarrow$	$x + (-y)$
	$--x$	$\rightarrow$	x
	$(-1) + 1$	$\rightarrow$	0
	$-(x + 1) + 1$	$\rightarrow$	$-x$
E:	$(x + y) + z$	$\rightarrow$	$x + (y + z)$

$\varphi(-) = 0$, $\varphi(1) > \varphi(0)$

$- \triangleright 0$

$- \triangleright +$