universität innsbruck

Seminar Report

AC Compatible Simplification Orders

Alexander Lochmann

8 March 2018

Supervisor: Dr. Christian Sternagel

Abstract

Many applications feature associative and commutative (AC) operators. Thus it is a point of interest to automatically proof termination of term rewrite systems with AC symbols. This can be achieved by finding an AC-compatible simplification order. This report recaps on existing AC-compatible simplification orders and it restates the analyzed results concerning their power and complexity. Additionally we reason which order is preferred to formalize in Isabelle/HOL.

Contents

1 Introduction

Many applications feature associative and commutative (AC) operators. As an example consider automated reasoning over algebraic structures such as groups and vector spaces. Therefore automatically generating termination proofs of term rewrite systems with AC symbols is a point of interest. This can be done by finding a suitable AC-compatible simplification order. Yamada et al. [6] analyzes the existing AC-compatible simplification orders. This report summarizes the complexity results. At the current state there is no formalization of any AC-compatible simplification orders in IsaFoR.

This report assumes general knowledge of term rewriting and is organized in the following way. First we introduce some preliminaries which are used in the report. Then recap the definitions of various AC-compatible simplification orders. Then look how they relate to each other. After that we discuss the complexity of the membership and the orientation problem of the orders and at the end we reason why we chose to formalize the $>_{\mathsf{ACKBO}}$ order.

2 Preliminaries

In this section we recap basic definitions from term rewrite systems. Additionally we introduce the notation and assumptions that are used in this report.

First we define basic notation and abbreviations. Then we extend the standard definitions of lexicographic and multiset orders to a more general one. After that we recap the definitions of simplification orders and then we introduce the concept of rewriting modulo equations and as last part we recall KBO.

A signature is a finite set of function symbols with associated arities. The signature of non AC symbols is denoted by F and the signature containing only binary AC symbols is denoted by F_{AC}. We denote the set of variables by V, the set of terms by $T(F \cup F_{AC}, V)$ and variables by x, y and z. Given a term s and a variable x, we denote the number of occurrences of x in s by $|s|_x$. The congruence relation $=_{AC}$ over the set of terms satisfies the following conditions $f(x, y) =_{AC} f(y, x)$ and $f(x, f(y, z)) =_{AC} f(f(x, y), z)$ for arbitrary terms x, y, z and $f \in F_{AC}$. A pair $(\succsim, \succ)$, where $\succ$ is a strict order and $\succsim$ a preorder, is an *order pair* if and only if $\succsim \cdot \succ \cdot \succsim \subseteq \succ$.

Let $\succ$ be a strict order and $\succsim$ a preorder on a set. We define the lexicographic extensions $\succ^{lex}$ and $\succsim^{lex}$ as follows

$$\begin{aligned}
(x_1, \ldots, x_m) \succsim^{lex} (y_1, \ldots, y_m) &\iff \forall i.\, 1 \leq i \leq m \implies x_i \succsim y_i \\
(x_1, \ldots, x_m) \succ^{lex} (y_1, \ldots, y_m) &\iff \exists k.\, x_k \succ y_k \land \forall i.\, (i < k \implies x_i \succsim y_i)
\end{aligned}$$

We define the multiset extensions $\succ^{mul}$ and $\succsim^{mul}$ as follows

$$\begin{aligned}
\{x_1, \ldots, x_m\} \succsim^{mul} \{y_1, \ldots, y_n\} &\iff m \geq n \land \forall i.\, (1 \leq i \leq n \implies x_i \succsim y_i) \\
\{x_1, \ldots, x_m\} \succ^{mul} \{y_1, \ldots, y_n\} &\iff \exists k.\, (k < n \land \forall i.\, (1 \leq i \leq k \implies x_i \succsim y_i) \land \\
&\qquad \forall j.\, (k < j \leq n \implies \exists l.\, l > k \land x_l \succ y_j))
\end{aligned}$$

A strict order on terms $\succ$ is a *rewrite relation* if it is closed under contexts and substitutions. Formally, given two terms s, t if $s \succ t$ then $C[s] \succ C[t]$ for all contexts C and $s\sigma \succ t\sigma$ for all substitutions σ.

A relation R on terms has the *subterm property* if $C[s]Rs$ holds for all non-empty contexts C and terms s.

2.1 Simplification orders

A rewrite relation R with the subterm property is called a *simplification order*.

Simplification orders are well founded by construction [2]. Hence if a TRS is compatible with a simplification order then it terminates.

2.2 Rewriting modulo equations

Consider a TRS R. We say that R terminates module E, a set of equations, if there exists no infinite rewrite sequence $t_1 \to_S t_2 \to_S t_3 \to_S \ldots$, where $\to_S = \to_E^* \cdot \to_R \cdot \to_E^*$, is infinite.

An AC-compatible simplification order $\succ$ is a simplification order where $(=_{AC}, \succ)$ is an order pair.

2.3 Weight functions

In this subsection we recall the definition of a weight function over $T(F \cup F_{AC}, V)$.

Given a signature F, a constant $w_0 > 0$ and a function $w : F \to N$ where for each constant $c \in F$ it follows that $w(c) \geq w_0$.

We extend the pair (w, w_0) to a weight function over $T(F \cup F_{AC}, V)$ as follows

$$w(t) := \left\{ \begin{array}{ll} w_0, & \text{if } t \in V \\ w(f) + \sum_{i=1}^{n} w(t_i) & \text{if } t = f(t_1, \ldots, t_n) \end{array} \right\}$$

Notice that we overload the symbol w. The functions can be distinguished by the provided argument.

Given a precedence $>$ over F. We call a weight function *admissible* if the following condition holds

$$\forall f. f \in F \land f \text{ is unary} \land w(f) = 0 \implies \forall g. (f \neq g \implies f > g)$$

2.4 The Knuth-Bendix Order KBO

In this subsection we recall the definition of the Knuth-Benedix order.

Let $>$ be a precedence and (w, w_0) a weight function. The order $>_{KBO}$ is inductively defined as follows: $s >_{KBO} t$ if $|s|_x \geq |t|_x$ for all $x \in V$ and either $w(s) > w(t)$ or $w(s) = w(t)$ and one of the following holds

1. $s = f^k(t)$, $t \in V$ for some $k > 0$

2. $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $f > g$

3. $s = f(s_1, \ldots, s_m)$, $t = f(t_1, \ldots, t_m)$, $(s_1, \ldots, s_m) >^{lex}_{KBO} (t_1, \ldots, t_m)$

3 ACKBO orders

In this section we first introduce some definitions which are used to define the AC-KBO orders. Then we will look at orders analyzed by Yamada et al[6]. The first order was published by Steinbach [5], the second one was published by Korovin and Voronkov[1] and at last we look at the definition of ACKBO by Yamada et al[6].

The *top-flattening* [3] of a term t with respect to an AC symbol f is the following multiset

$$\nabla_f(t) := \left\{ \begin{array}{ll} \{t\}, & \text{if } root(t) \neq f \\ \nabla_f(t_1) \uplus \nabla_f(t_2), & \text{if } t = f(t_1, t_2) \end{array} \right\}$$

Example 3.1.

$$\nabla_+(a + b + a) = \{a, a, b\}$$

Given a multiset of terms T, we define the submultiset $T \restriction_v$ of T by

$$T \restriction_v := \{x \in T \mid x \in V\}$$

and given a function symbol f and a binary relation R on a signature we have

$$T \restriction^{Rf} := \{t \in T \setminus V \mid root(t)Rf\}$$

Example 3.2. Consider the multiset $T = \{f(b), g(a), g(x), x, y\}$ and the precedence $f > + > g$, then we have $T \restriction_v = \{x, y\}$ and $T \restriction^{\not< +} := \{f(b)\}$.

3.1 Steinbach's order

Seinbach published the AC-compatible KBO $>_s$. It is defined as follows

Definition 3.3. Let $>$ precedence and (w, w_0) be an admissible weight function. The order $>_s$ is inductively defined as follows: $s >_{KBO} t$ if $|s|_x \geq |t|_x$ for all $x \in V$ and either $w(s) > w(t)$ or $w(s) = w(t)$ and one of the following holds

1. $s = f^k(t)$, $t \in V$ for some $k > 0$

2. $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $f > g$

3. $s = f(s_1, \ldots, s_m)$, $t = f(t_1, \ldots, t_m)$, $f \notin F_{AC}$, $(s_1, \ldots, s_m) >^{lex}_s (t_1, \ldots, t_m)$

4. $s = f(s_1, s_2)$, $t = f(t_1, t_2)$, $f \in F_{AC}$, $S = \nabla_f(s)$, $T = \nabla_f(t)$, $S >^{mul}_s T$

Here $=_{AC}$ is used as preorder in $>^{lex}_s$ and $>^{mul}_s$.

The only difference to KBO is the restriction to non AC function symbols in the second case and the additional case 4 for AC symbols.

Example 3.4. Example TRS R over signature $F = \{f, a\}$, $F_{AC} = \{+\}$ and consisting of the rules

$$f(a + a) \to f(a) + f(a) \qquad\qquad a + f(f(a)) \to f(a) + f(a)$$

We choose the precedence $f > a > +$ and the weights $w(+) = w(a) = 1$ and $w(f) = 0$. Then the first rule is oriented by case 2. For the second rule we apply the top-flattening and we get the multisets

$$S = \{a, f(f(a))\} \qquad T = \{f(a), f(a)\}.$$

The order $>_s$ has the subterm property hence $f(f(a)) >_s f(a)$.

Following theorem was proven by Steinbach [5].

Theorem 3.5. *If every symbol in F_{AC} is minimal with respect to $>$ then $>_s$ is an AC-compatible simplification order.*

3.2 Korovin and Voronkov's order

Korovon and Voronkov introduce several orders [1]. The most general one is not closed under context as shown by Yamada et al[6, Example 4.8]. Therefor we will only recall an extended version which is closed under contexts.

We first define two auxiliary relation $\geq_{kvau'}$ and $>_{kvau'}$ on terms.

Definition 3.6. $s \geq_{kvau'} t$ if $|s|_x \geq |t|_x$ for all $x \in V$ and either $w(s) > w(t)$ or $w(s) = w(t)$ and either $root(s) \geq root(t)$ or $t \in V$

$s >_{kvau'} t$ if $|s|_x \geq |t|_x$ for all $x \in V$ and either $w(s) > w(t)$ or $w(s) = w(t)$ and $root(s) > root(t)$

Example 3.7. Let c be a constant and f an unary symbol. If $w(f) = 0$ then admissibility imposes $f > c$. So $f(c) >_{kvau'} c$ but $f(x) >_{kvau'} x$ does not hold.

Definition 3.8. Let $>$ precedence and (w, w_0) be an admissible weight function. The order $>_{KV'}$ is inductively defined as follows: $s >_{KV'} t$ if $|s|_x \geq |t|_x$ for all $x \in V$ and either $w(s) > w(t)$ or $w(s) = w(t)$ and one of the following holds

1. $s = f^k(t)$, $t \in V$ for some $k > 0$
2. $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $f > g$
3. $s = f(s_1, \ldots, s_m)$, $t = f(t_1, \ldots, t_m)$, $f \notin F_{AC}$, $(s_1, \ldots, s_m) >^{lex}_{KV'} (t_1, \ldots, t_m)$
4. $s = f(s_1, s_2)$, $t = f(t_1, t_2)$, $f \in F_{AC}$, $S = \nabla_f(s)$, $T = \nabla_f(t)$,
   a) $S \restriction^{\not< f} >^{mul}_{kvau'} T \restriction^{\not< f} \uplus T \restriction_v - S \restriction_v$ or

b) $S \upharpoonright^{\not< f} \geq^{mul}_{kvau'} T \upharpoonright^{\not< f} \uplus T \upharpoonright_v - S \upharpoonright_v$, $|S| > |T|$ or

c) $S \upharpoonright^{\not< f} \geq^{mul}_{kvau'} T \upharpoonright^{\not< f} \uplus T \upharpoonright_v - S \upharpoonright_v$, $|S| = |T|$ and $S >^{mul}_{KV'} T$

Here $\geq_{kvau'}$ is used as pre order in case 4a and $=_{AC}$ is used as pre order in $S >^{mul}_{KV'} T$. The recursive call only occurs in the case 3 and 4c.

Example 3.9. Example TRS R over signature $F = \{f, c\}$, $F_{AC} = \{+\}$ and consisting of the rules

$$f(x) \to x \qquad\qquad f(x) + y \to x + y$$

We choose the precedence $f > + > c$ and the weights $w(+) = w(c) = 1$ and $w(f) = 0$. From case 1 we have that $f(x) >_{KV'} x$. For the second rule we need to apply the top-flattening and we get the multisets

$$S = \{f(x), y\}\, T = \{x, y\}.$$

We have $S \upharpoonright^{\not< +} = \{f(x)\}$ and $T \upharpoonright^{\not< +} \uplus T \upharpoonright_v - S \upharpoonright_v = \{x\}$. From $f(x) \geq_{kvau'} y$ case 4c applies because $f(x) \geq^{mul}_{kvau'} y$, $|S| = |T| = 2$ and $S = \{f(x), y\} \geq_{KV'} \{x, y\} = T$.

Following theorem was proven by Yamada et al [6].

Theorem 3.10. *The order $>_{KV'}$ is an AC-compatible simplification order.*

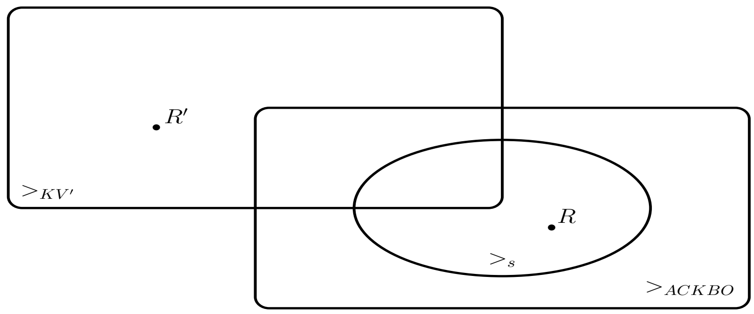
From the inclusion $>_{KV} \subseteq >_{KV'}$ shown by [6], it follows that $>_{KV}$ is a sound method for proving termination of TRS, with AC symbols. Here $>_{KV}$ denotes the original version of Korovin and Voronkov's order as defined by [6].

3.3 AC-KBO

Here we recall the order introduced by Yamada et al [6].

Definition 3.11. Let $>$ precedence and (w, w_0) be an admissible weight function. The order $>_{ACKBO}$ is inductively defined as follows: $s >_{ACKBO} t$ if $|s|_x \geq |t|_x$ for all $x \in V$ and either $w(s) > w(t)$ or $w(s) = w(t)$ and one of the following holds

1. $s = f^k(t)$, $t \in V$ for some $k > 0$
2. $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, $f > g$
3. $s = f(s_1, \ldots, s_m)$, $t = f(t_1, \ldots, t_m)$, $f \notin F_{AC}$, $(s_1, \ldots, s_m) >^{lex}_{ACKBO} (t_1, \ldots, t_m)$
4. $s = f(s_1, s_2)$, $t = f(t_1, t_2)$, $f \in F_{AC}$, $S = \nabla_f(s)$, $T = \nabla_f(t)$,
   a) $S \upharpoonright^{\not< f} >^{mul}_{ACKBO} T \upharpoonright^{\not< f} \uplus T \upharpoonright_v - S \upharpoonright_v$ or
   b) $S \upharpoonright^{\not< f} =^{mul}_{AC} T \upharpoonright^{\not< f} \uplus T \upharpoonright_v - S \upharpoonright_v$, $|S| > |T|$ or
   c) $S \upharpoonright^{\not< f} =^{mul}_{AC} T \upharpoonright^{\not< f} \uplus T \upharpoonright_v - S \upharpoonright_v$, $|S| = |T|$, $S \upharpoonright^{< f} >^{mul}_{ACKBO} T \upharpoonright^{< f}$



Here $=_{AC}$ is used as preorder in $>^{lex}_{ACKBO}$ and $>^{mul}_{ACKBO}$.

Following theorems were proven by Yamada et al [6].

Theorem 3.12. *The order $>_{ACKBO}$ is an AC-compatible simplification order.*

Theorem 3.13. *If every AC symbol has minimal precedence then $>_s = >_{ACKBO}$.*

4 Relation between ACKBO orders

In this section we look at the relation between the orders and we will show with an example that the orders $>_{KV'}$ and $>_{ACKBO}$ are distinct.

In Figure 4 we see the relation between the orders where the example TRS R consists of the rules

$$f(a+a) \to f(a) + f(a) \qquad\qquad a + f(f(a)) \to f(a) + f(a)$$

and we obtain the TRS R' by inverting the second rule. We show that R can be oriented with $>_{ACKBO}$ but not with $>_{KV'}$, here $+ \in F_{AC}$.

To orient the first rule it is necessary that $w(f) = 0$, because the symbol f occurs twice on the right-hand side and the occurrences of the other symbols don't change. Admissibility requires that $f > a$ and $f > +$ in the precedence. Therefore rule 1 is oriented by both orders. For the second rule we have the top-flattening of the terms $S = \{a, f(f(a))\}$ and $T = \{f(a), f(a)\}$. Now we have two possible cases for the precedence

case 1: $a > +$ then

$$S \restriction^{\not< +} = \{a, f(f(a))\}\, T \restriction^{\not< +} = \{f(a), f(a)\}$$

For $>_{ACKBO}$ we have that $f(f(a)) >_{ACKBO} f(a)$, hence the rule can be oriented by case 4a. For $>_{KV'}$ we have that $f(f(a)) \not\geq_{kvau'} f(a)$, so case 4a does not apply and $a \not\geq_{kvau'} f(a)$ hence case 4b and 4c does also not apply.

case 2: $+ > a$ then

$$S \restriction^{\not< +} = \{f(f(a))\}\, T \restriction^{\not< +} = \{f(a), f(a)\}$$

We see that the argumentation of the previous case still holds for both orders.

method	membership	orientability
Steinbach	P	?
ACKBO	P	NP-complete
KV	P	NP-complete
KV'	NP-complete	NP-complete

Figure 1: Complexities of the orders taken from [6].

Now lets consider inverting the second rule then we have in case 1 the following multisets

$$S \restriction^{\not< +} = \{f(a), f(a)\}\, T \restriction^{\not< +} = \{a, f(f(a))\}$$

For $>_{KV'}$ we have that $f(a) \geq_{kvau'} f(f(a))$ and $f(a) >_{kvau'} a$, hence the rule can be oriented by case 4a. For $>_{ACKBO}$ we have that $f(a) \neq_{AC} f(f(a))$ and $f(a) \ngtr_{ACKBO} f(f(a))$, hence no rule applies. Analog for the 2 case.

case 2: $+ > a$ then

$$S \restriction^{\not< +} = \{f(a), f(a)\}\, T \restriction^{\not< +} = \{f(f(a))\}$$

We see that the argumentation of the previous case still holds for both orders.

5 Complexity

In this section we recall the membership and the orientability problem. An overview of the complexities of the different algorithms is shown in 1. Here the symbol P denotes that the problem can be solved in polynomial time and NP for nondeterministic polynomial time. We will only look at the membership proof for $>_{ACKBO}$ and present a proof sketch for the orientability problem.

For the membership problem we have a given precedence and admissible weight function and we check if $s >_{ACKBO} t$ holds.

For the orientability problem we need to check if there exists a precedence and a weight function such that for a given TRS R it holds that $R \subseteq >_{ACKBO}$.

Yamada et al [6] state the conjecture that the orientability of Steinbach's order can be checked in polynomial time.

5.1 Membership

In this subsection we prove that $>_{ACKBO}$ membership can be checked in polynomial time. For this we split the proof in three parts. First we prove two lemmas which show that under certain assumptions multiset extension preserves polynomial complexity.

Lemma 5.1. *Let $(\succsim, \succ)$ be an order pair and $\sim := \succsim \setminus \succ$ be symmetric. If $s \sim t$ then $M \succ^{mul} N$ and $M \uplus \{s\} \succ^{mul} N \uplus \{t\}$ are equivalent.*

We prove the inclusion from right to left, bacause that is the part of the lemma used in the membership proof.

Proof. There exist indices i, j such that $s_i = s$ and $t_j = t$. Let k be the number of elements in the preorder case.

1. Case $i, j \leq k$

   Then $s_j \gtrsim t_j = t \sim s = s_i \gtrsim t_i$. From transitivity it follows that $s_j \gtrsim t_i$. Remove s_j and t_j from multisets then replace s_i with s_j.

2. Case $i \leq k < j$

   Then $s_l \succ t_j = t \sim s = s_i \gtrsim t_i$ and $l > k$. From the definition of order pair it follows that $s_l \succ t_i$. Remove s_i and t_i then replace t_j with t_i.

3. Case $j \leq k < i$

   Then $s_j \gtrsim t_j = t \sim s = s_i \succ t_l$ for some $l > k$. From the definition of order pair it follows that $s_j \succ t_l$ for all l which satisfy $s_i \succ t_l$. Swap s_j with s_i then remove s_i and t_j.

4. Case $k < j, i$

   Then $s_j \succ t_j = t \sim s = s_i \succ t_i$. From the definition of order pair it follows that $s_j \succ t_l$ for all l which satisfy $s_i \succ t_l$. Remove s_i and t_j.

□

Lemma 5.2. *Let $(\gtrsim, \succ)$ be an order pair and $\sim := \gtrsim \setminus \succ$ be symmetric. If the decision problem for $\gtrsim$ and $\succ$ are in P then the decision problem for $\succ^{mul}$ is in P.*

Proof. Consider the multisets S and T. To check if $S >^{mul} T$ we perform the following steps

For each $(s, t) \in S \times T$ check if $s \sim t$. If thats the case then remove the elements s, t from the corresponding multiset. From the previous lemma it follows that this transformation preserves the result.

Then check if for each $t \in T$ there is on $s \in S$ such that $s \succ t$.

□

KBO membership can be checked in polynomial time and the base cases of $> ACKBO$, rules 1 - 3, are equal to KBO. From both lemmas and the induction hypothesis membership in polynomial time follows.

5.2 Orientation

The problem of the orientation proof is that we need to consider infinitely many weight functions. Therefore we won't guess the weight of the symbols but rather the relation between the weights of the symbols. For example if we have two function symbols f, g then there are three possibilities $f > g$, $f < g$ or $f \geq g \wedge g \geq f$.

Constructing all possible inequalities between the function symbols reduces the search space of the weight function to a finite one.

Now consider a TRS R, then we collect all the terms that occur in the rules $R_t := \{t \mid \exists u(tRu \vee uRt)\}$. We replace each variable in term $t \in R_t$ with a fresh constant w_0. Then we will take all subterms of these terms $S := \{t \mid s \in R_t \wedge s \trianglerighteq t\}$. The set S contains all terms relevant to orient R. If S has at least two elements then the set of all possible inequalities has the size of $3^{|S|-1}$, note that $s = t$ can be represented as inequality ($s \geq t$ and $t \geq s$).

For each set of inequalities we check if there exists a precedence so that R can be oriented and the set of inequalities has a solution. The set of possible precedences is finite, because signatures are finite.

From the previous section we know that the membership problem is in P. Finding a solution of a set of inequalities in $\mathbb{R}$ is in P [4]. Note that finding a solution in $\mathbb{Z}$ is in NP, but we only need to ensure that a set has a solution and for each solution in $\mathbb{R}$ there exists a solution in $\mathbb{Z}$.

Hence orientability is NP, because the search space grows exponentially.

6 Further work

The aim is to formalize an AC-compatible simplification order in IsaFoR. Additionally program a certifier in Isabelle and show that the program is compatible to the formalized order. The certifier should verify that for a given TRS R, weight function and precedence it holds that R is a subset of the formalized AC-compatible simplification order. At the present state there is no formalization of an AC-compatible simplification order in IsaFoR. Therefore we recap the results of this report and reason which order is suited to formalize.

As we saw in figure 1 the membership problem in $>_{KV'}$ is NP-complete. The certifier must check if each rule is oriented from left to right. Therefore the membership problem of the checked order should be in P. Thus the other orders are preferable over $>_{KV'}$.

From Theorem 3.13 we know that $>_s \subseteq >_{ACKBO}$ and $>_{KV}$ is not AC-compatible simplification order. Therefore it is reasonable to choose $>_{ACKBO}$ to be formalized in IsaFoR.

References

[1] Konstantin Korovin and Andrei Voronkov. “Orienting rewrite rules with the Knuth-Bendix order”. In: *Information and Computation* 183.2 (2003). 12th International Conference on Rewriting Techniques and Applications (RTA 2001), pp. 165 –186. ISSN: 0890-5401. DOI: `https://doi.org/10.1016/S0890-5401(03)00021-X`.

[2] Aart Middeldorp and Hans Zantema. “Simple Termination of Rewrite Systems”. In: 175 (Mar. 1997), pp. 127–158.

[3] Albert Rubio. “A Fully Syntactic AC-RPO”. In: *Information and Computation* 178.2 (2002), pp. 515 –533. ISSN: 0890-5401. DOI: `https://doi.org/10.1006/inco.2002.3158`.

[4] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, Chichester, 1986.

[5] Joachim Steinbach. “AC-Termination of rewrite systems: A modified Knuth-Bendix ordering”. In: *Algebraic and Logic Programming*. Ed. by Hélène Kirchner and Wolfgang Wechler. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 372–386. ISBN: 978-3-540-46738-0.

[6] Akihisa Yamada, Sarah Winkler, Nao Hirokawa, and Aart Middeldorp. “AC-KBO revisited”. In: *Theory and Practice of Logic Programming* 16.2 (2016), pp. 163–188. DOI: `10.1017/S1471068415000083`.