AC-KBO Revisited

Akihisa Yamada (Nagoya U. / AIST)
Sarah Winkler (University of Innsbruck)
Nao Hirokawa (JAIST)
Aart Middeldorp (University of Innsbruck)

at FLOPS 2014
Background

**Completion** [Knuth & Bendix '70]:

Given axioms

\[ E = \begin{cases} 
  x & \leftarrow x + 0 \\
  s(x) + y & \leftarrow s(x + y) \\
  x + (y + z) &= (x + y) + z \quad (A) \\
  x + y &= y + x \quad (C) 
\end{cases} \]

Generate an equivalent convergent TRS.

**Solution** (in very essence):

Orient equations by a *simplification order*, e.g.

- *Knuth-Bendix Order (KBO)* [Knuth & Bendix '70]
- *Recursive Path Order (RPO)* [Dershowitz '82]

Neither is simpler!
**Background**

**AC-Completion** [Peterson & Stickel '86]:

Given axioms

\[ E = \begin{cases} 
  x \leftarrow x + 0 \\
  s(x) + y \rightarrow s(x + y) \\
  x + (y + z) = (x + y) + z \quad (A) \\\n  x + y = y + x \quad (C)
\end{cases} \]

Generate an equivalent convergent **AC-TRS**.

**Solution** (in very essence):

Orient \( E \setminus \text{AC} \) by an **AC-simplification order**, e.g.

- **AC-KBO** [Steinbach '90, Korovin & Voronkov '03]
- **AC-RPO** [..., cf. Rubio '02]

---

The topic

A. Yamada, S. Winkler, N. Hirokawa, A. Middeldorp: AC-KBO Revisited
History of AC-KBO

- First AC-KBO [Steinbach '90] ...
  → Claimed to be incorrect [Korovin & Voronkov '03]

- Second AC-KBO [KV '03]
  - Correctness proof is omitted
  - Not (reportedly) implemented / experimented
  → so we tried, and found... it's still incorrect!
  ...but sound (we extend it to a correct one)

- Third AC-KBO [this work]
  - Correct, implemented (in TTT2), experimented
  - Subsumes [Steinbach '90]. Thus...
Standard KBO

$s >_{\text{KBO}} t$ iff

- $w(s) > w(t)$; or $w(s) = w(t)$ and
  - (0) $s = f(\ldots f(t)\ldots)$, or
  - (1) $s = f(\ldots)$, $t = g(\ldots)$, $f > g$, or
  - (2) $s = f(s_1, \ldots)$, $t = f(t_1, \ldots)$, $(s_1, \ldots) >_{\text{lex}}^{\text{KBO}} (t_1, \ldots)$.

- Where $w$ assigns a 'weight' to a term, and
- $>$ specifies 'precedence' on function symbols

**Theorem** [cf. Baader & Nipkow '98]

$>_{\text{KBO}}$ is a simplification order.
AC-KBO by [Steinbach '90]

\[ s \succ_s t \text{ iff} \]

- \( w(s) > w(t) \); or \( w(s) = w(t) \) and
  - \( s = f(\ldots f(t)\ldots), \) or
  - \( s = f(\ldots), t = g(\ldots), f > g, \) or
  - \( s = f(s_1,\ldots), t = f(t_1,\ldots), (s_1,\ldots) >_{\text{lex}} s (t_1,\ldots), \) or
  - \( s = s_1 \ast \cdots \ast s_n, t = t_1 \ast \cdots \ast t_m, \) and
    \[ \{s_1,\ldots,s_n\} >_{s}^{\text{mul}} \{t_1,\ldots,t_m\} \]

special case for AC symbols

**Theorem** [Steinbach '90]

If \( \ast \) is minimal w.r.t. \( > \) and \( w(\ast) = 0 \), then \( >_s \) is an AC-simplification order.
AC-KBO by [Korovin & Voronkov '03]

\[ s \succ_{KV} t \text{ iff} \]

- \( w(s) > w(t); \) or \( w(s) = w(t) \) and
  - (0) \( s = f(\ldots f(t)\ldots), \) or
  - (1) \( s = f(\ldots), t = g(\ldots), f > g, \) or
  - (2) \( s = f(s_1,\ldots), t = \cdot \cdot \cdot \)
    - (3) \( s = s_1 \ast \cdots \ast s_n, t = t_1 \ast \cdots \ast t_m, \)
      - \{s_1,\ldots,s_n\} \succ^\text{mul}_{KV} \{t_1,\ldots,t_m\} \uparrow, \) or
      - \{s_1,\ldots,s_n\} \equiv^\text{mul}_{KV} \{t_1,\ldots,t_m\} \uparrow, \) and
        - \( n > m, \) or
        - \( n = m \) and \( s_1,\ldots,s_n \succ^\text{mul}_{KV} \{t_1,\ldots,t_m\}. \)

extended using an idea of AC-RPO [Rubio '02]...
Auxiliary order $>_\text{kv}$

$s >_\text{kv} t$ iff

- $w(s) > w(t)$; or $w(s) = w(t)$ and
  - $s = f(...f(t)...)$, (0) $s = f(...)$, $t = g(...)$, $f > g$.
  - $s = f(s_1,...)$, $t = f(t_1,...)$, $(s_1,...) >_{\text{lex}} (t_1,...)$, (2)
  - $s = s_1 * ... * s_n$, $t = t_1 * ... * t_m$, $s_1 >_{\text{mul}} t_1$, ..., $s_n >_{\text{mul}} t_m$, (3)

$>_\text{KV}$ is not an AC-simplification order!

... please see the paper for a counterexample.
Rescuing $>_{KV}$

$s >_{KV'} t$ iff

- $w(s) > w(t)$; or $w(s) = w(t)$ and
  - (0) $s = f(\cdots f(t)\cdots)$,
  - (1) $s = f(\ldots), t = g(\ldots), f > g$.

**Theorem**

Define $>_{KV'}$ using $\geq_{KV'}$ instead of $=_{KV}$. Then $>_{KV'}$ is an AC-simplification order.

**Proof**: far from trivial... Cf. the full version.

**Corollary**: $>_{KV} \subseteq >_{KV'}$. Thus $>_{KV}$ is sound anyway.
Our AC-KBO

\[ s >_{\text{ACKBO}} t \text{ iff } \]

1. \( w(s) > w(t); \) or \( w(s) = w(t) \) and
2. \( s = f(\cdots f(t)\cdots), \)
3. \( s = f(s_1, \ldots), t = g(t_1, \ldots), \)
4. \( s = s_1 \cdot \cdots \cdot s_n, t = t_1 \cdot \cdots \cdot t_m, \)
   - \( \{s_1, \ldots, s_n\} \uparrow_{\text{ACKBO}} \{t_1, \ldots, t_m\} \uparrow, \)
   - \( \{s_1, \ldots, s_n\} \uparrow =_{\text{mul}} \{t_1, \ldots, t_m\} \uparrow_{\text{mul}} \) and

**Theorem**

\( >_{\text{ACKBO}} \) is an AC-simplification order.

**Proof:** slightly better... Please see the proceedings.

A. Yamada, S. Winkler, N. Hirokawa, A. Middeldorp: AC-KBO Revisited
Correctness of [Steinbach '90]

**Theorem**
If the precondition for Steinbach's theorem holds, then $\succ_{\text{ACKBO}} = \succ_S$

**Corollary**
$\succ_S$ is a correct AC-simplification order.
Comparison

- Corrected KV' New AC-KBO
- Wrong KV
- Corrected
- Wrong Steinbach
Complexity results
The orientability problem

**Problem (Orientability)**

Given \( l_1 \rightarrow r_1 \) \( \vdots \) find an AC-KBO s.t. \( l_n \rightarrow r_n \)

**Theorem**

AC-KBO orientability is NP-hard.

**Proof:**

By reduction from a CNF satisfiability problem.

**Note:** KBO orientability is P [Korovin & Voronkov '03]
AC-KBO again

\[ s >_{\text{ACKBO}} t \text{ iff} \]

- \( w(s) > w(t) \); or \( w(s) = w(t) \) and
  
  (0) \( s = f^k(t) \),
  
  (1) \( s = f(s_1, \ldots), t = g(t_1, \ldots), f > g \),
  
  (2) \( s = f(s_1, \ldots), t = f(t_1, \ldots), (s_1, \ldots) >_{\text{ACKBO}}^\text{lex} (t_1, \ldots) \),
  
  (3) \( s = s_1 \ast \cdots \ast s_n, t = t_1 \ast \cdots \ast t_m \),

- \( \{s_1, \ldots, s_n\} \uparrow_* >_{\text{ACKBO}}^\text{mul} \{t_1, \ldots, t_m\} \uparrow_* \), or
  
  \( \{s_1, \ldots, s_n\} \uparrow_* =_{\text{AC}}^\text{mul} \{t_1, \ldots, t_m\} \uparrow_* \), and

  - \( n > m \), or
  - \( n = m \) and \( \{s_1, \ldots, s_n\} >_{\text{ACKBO}}^\text{mul} \{t_1, \ldots, t_m\} \).
The operation $\uparrow_*$

- A high-pass filter w.r.t. precedence called BigHead/NoSmallHead in AC-RPO [Rubio '02]

- Example:

  if $f > g > h > i > j$ then
  \[
  \{f(x), g(x), i(x), j(x)\} \uparrow_h = \{f(x), g(x)\}
  \]

Let us assume $\text{True} > * > \text{False}$. Then,
\[
\{T, T, F, F, T\} \uparrow_* = \{T, T, T\}
\]
Encoding positive literals

- Consider $p_1 \lor p_2 \lor p_3 \lor \neg p_4 \lor \neg p_5 \lor \neg p_6$
- We have

\[
\{ \square \square \square \} > \text{mul} \{ \blu, \grn, \yel, \ldots \}
\]
Encoding positive literals

- Consider $p_1 \lor p_2 \lor p_3 \lor \neg p_4 \lor \neg p_5 \lor \neg p_6$

- But

\[
\{ \square, \square, \square \} \not\overset{\text{mul}}{\upharpoonright} \{ \square, \square, \square, \ldots \}
\]

- Thus, $p_1 \lor p_2 \lor p_3$ iff

\[
\{ p_1(\square), p_2(\square), p_3(\square) \} \upharpoonright_* >_{\text{mul}} \{ \square, \square, \square, \ldots \}
\]

by choosing big enough \square
Encoding negative literals

- Consider $p_1 \lor p_2 \lor p_3 \lor \neg p_4 \lor \neg p_5 \lor \neg p_6$
- We have

\[
\left\{ \begin{array}{c}
\text{\textcolor{red}{Red}} \\
\text{\textcolor{blue}{Blue}} \\
\text{\textcolor{green}{Green}} \\
\text{\textcolor{orange}{Orange}}
\end{array} \right\} \Rightarrow \text{mul} \left\{ \begin{array}{c}
\text{\textcolor{red}{Red}} \\
\text{\textcolor{blue}{Blue}} \\
\text{\textcolor{green}{Green}} \\
\text{\textcolor{orange}{Orange}}
\end{array} \right\}
\]
Encoding negative literals

- Consider \( p_1 \lor p_2 \lor p_3 \lor \neg p_4 \lor \neg p_5 \lor \neg p_6 \)
- But,

\[
\{ p_4 (\square), p_5 (\square), p_6 (\square) \}\uparrow^* >^\text{mul} \{ p_4 (\square), p_5 (\square), p_6 (\square) \}\uparrow^*
\]

Thus, \( \neg p_4 \lor \neg p_5 \lor \neg p_6 \) iff

- with choosing big enough \( \square, \square, \) or \( \square \)
Complexity results

- ...this way we reduce CNF satisfiability
  - although some more tricks are needed to adjust to true AC-KBO orientability
  - The same idea works for $>_{KV}$
- For $>_{KV'}$, even testing $s >_{KV'} t$ is NP-hard

**Note:** It is linear for standard KBO [Löchner '06]

<table>
<thead>
<tr>
<th></th>
<th>KBO</th>
<th>Steinbach</th>
<th>AC-KBO</th>
<th>KV</th>
<th>KV'</th>
</tr>
</thead>
<tbody>
<tr>
<td>membership</td>
<td>Linear [Löchner '06]</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>NP-hard</td>
</tr>
<tr>
<td>orientability</td>
<td>P [KV '03]</td>
<td>?</td>
<td>NP-hard</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>
An extension, implementation, and experiments
Transfinite KBO [Ludwig & Waldmann'07]

- Weight by linear ordinal polynomial:
  \( w(x) = x, \)
  \( w(f(s_1, \ldots, s_n)) = w_f \oplus c_{f,1} \odot w(s_1) \oplus \ldots \oplus c_{f,n} \odot w(s_n) \)

- [Winkler+ '12] showed...
  - Ordinals are needless (for finite TRSs).
  - Significant power-up over KBO.
TKBO with AC

- Coefficients for AC symbols must be 1.

Conditions for AC-simplification orders
- Subterm property
- Stability
- Monotonicity
- **AC-compatibility**: \( s =_{AC} t > u =_{AC} v \Rightarrow s > v \)

\[ w(x \ast (y \ast z)) = w((x \ast y) \ast z) \] should hold, i.e.,
\[
\begin{align*}
(1 + c_{*,2})w_* + c_{*,1}x + c_{*,2} c_{*,1} y + (c_{*,2})^2 z \\
= (1 + c_{*,1})w_* + (c_{*,1})^2 x + c_{*,1} c_{*,2} y + c_{*,2} z
\end{align*}
\]
\[
c_{*,1} = 1 \quad c_{*,2} = 1
\]
Implementation & experiments

- **Implementation:**
  - Implemented in the Tyrolean Termination Tool 2 with help of SAT/SMT solvers incorporated into AC-DP method [Alarcón+ '10]
  - Combined with the AC-completion tool 'mascott'

- **Test set:** 145 AC-TRSs from
  - the Termination Problem Database (TPDB)
  - and literatures

- **Environment:**
  - 8x AMD Opteron 885 @2.6MHz, 64GB memory
Experimental results

<table>
<thead>
<tr>
<th>method</th>
<th>orientability</th>
<th>AC-DP</th>
<th>AC-completion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>yes</td>
<td>time</td>
<td>T.O.</td>
</tr>
<tr>
<td>Steinbach</td>
<td>23</td>
<td>1.6</td>
<td>0</td>
</tr>
<tr>
<td>KV</td>
<td>30</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>KV'</td>
<td>30</td>
<td>2.1</td>
<td>0</td>
</tr>
<tr>
<td>AC-KBO</td>
<td>32</td>
<td>1.7</td>
<td>0</td>
</tr>
<tr>
<td>+coefficient</td>
<td>37</td>
<td>47.1</td>
<td>0</td>
</tr>
<tr>
<td>AC-RPO</td>
<td>63</td>
<td>2.8</td>
<td>0</td>
</tr>
</tbody>
</table>

- $>_S$ to $>_KV$ is a big improvement.
- $>_KV$ to $>_KV'$ is merely a bugfix.
- $>_{ACKBO}$ is slightly more efficient than $>_KV'$
- AC-RPO is good, but AC-KBO is complementary
Conclusion

- (In)correctness of existing variants:
  - [Steinbach '90] is correct (while believed not)
  - [Korovin & Voronkov '03] is not a correct AC-simplification order, but sound

- New AC-KBOs:
  - A simpler (correct) version
  - extension with 'subterm coefficient'

- Complexity results
  - ... AC-KBO is much harder than the plain KBO

... Thank you for attention!