AC-KBO Revisited

at FLOPS 2014

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

Background

Completion [Knuth & Bendix '70]:

Given axioms

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

Generate an equivalent convergent T

i.e., fu

Neither is simpler!

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]

Background

AC-Completion [Peterson & Stickel '86]:

Given axioms
$$E = \begin{cases} x \leftarrow x + 0 \\ s(x) + y \rightarrow s(x + y) \\ x + (y + z) = (x + y) + z & (A) \\ x + y = y + x & (C) \end{cases}$$

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

Solution (in very essence):

Orient $E \setminus$ **AC** by an **AC**-*simplification order, e.g.*

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

The topic

History of AC-KBO

- First AC-KBO [Steinbach '90] **... correct!**
  - → ~~**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

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

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

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

Theorem [cf. Baader & Nipkow '98]
$>_{\mathrm{KBO}}$ is a simplification order.

AC-KBO by [Steinbach '90]

$s >_{\mathsf{S}} t$ iff

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

special case for AC symbols

Theorem [Steinbach '90]
If $*$ is minimal w.r.t. $>$ and $w(*) = 0$, then $>_{\mathsf{S}}$ is an AC-simplification order.

AC-KBO by [Korovin & Voronkov '03]

$s >_{\mathbf{KV}} t$ iff

- $w(s) > w(t)$; or $w(s) = w(t)$ and
  - **(0)** $s = f(\cdots f(t) \cdots)$, or
  - **(1)** $s = f(\dots)$, $t = g(\dots)$, $f > g$, or
  - **(2)** $s = f(s_1, \dots)$, $t =$ …
  - **(3)** $s = s_1 * \cdots * s_n$, $t = t_1$ …
    - $\{s_1, \dots, s_n\} \restriction_* >^{\mathrm{mul}}_{\mathbf{kv}} \{t_1, \dots, t_m\} \restriction_*$, or $\{s_1, \dots, s_n\} \restriction_* =^{\mathrm{mul}}_{\mathbf{kv}} \{t_1, \dots, t_m\} \restriction_*$, and
      - $n > m$, or
      - $n = m$ and $\{s_1, \dots, s_n\} >^{\mathrm{mul}}_{\mathbf{KV}} \{t_1, \dots, t_m\}$.

they use an auxiliary order

extended using an idea of AC-RPO [Rubio '02]…

Auxiliary order $>_{\mathrm{kv}}$

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

- $w(s) > w(t)$; or $w(s) = w(t)$ and
  - (0) $s = f(\cdots f(t) \cdots)$,
  - **(1)** $s = f(\ldots), t = g(\ldots), f > g$.
  - (2) $s = f(s_1, \ldots), t = f(t_1, \ldots), (s_1, \ldots) >^{\mathrm{lex}}_{\mathrm{KV}} (t_1, \ldots)$,
  - (3) $s = s_1 * \cdots * s_n, t = t_1 * \cdots * t_m$,
    - $\{s_1, \ldots, s_n\} \restriction_* >^{\mathrm{mul}}_{\mathrm{kv}} \{t_1, \ldots, t_m\} \restriction_*$, or
    - $\{s_1, \ldots, s_n\} \restriction_* =^{\mathrm{mul}}_{\mathrm{kv}} \{t_1, \ldots, t_m\} \restriction_*$ and

cannot be omitted!

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

... please see the paper for a counterexample.

Rescuing $>_{\mathrm{KV}}$

$s \geq_{\mathrm{kv}'} t$ 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 $>_{\mathrm{KV}'}$ using $\geq_{\mathrm{kv}'}$ instead of $=_{\mathrm{kv}}$.
Then $>_{\mathrm{KV}'}$ is an AC-simplification order.

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

Corollary: $>_{\mathrm{KV}} \subseteq >_{\mathrm{KV}'}$. Thus $>_{\mathrm{KV}}$ is sound anyway.

Our AC-KBO

$s >_{\mathrm{ACKBO}} t$ iff

- $w(s) > w(t)$; or $w(s) = w(t)$ and
  - **(0)** $s = f(\cdots f(t) \cdots)$,
  - **(1)** $s = f(s_1, \ldots)$, $t = g$
  - **(2)** $s = f(s_1, \ldots)$, $t = f$
  - **(3)** $s = s_1 * \cdots * s_n$, $t = t_1 * \cdots * t_m$,
    - $\{s_1, \ldots, s_n\}\restriction_* >^{\mathrm{mul}}_{\mathrm{ACKBO}} \{t_1, \ldots, t_m\}\restriction_*$, or
    - $\{s_1, \ldots, s_n\}\restriction_* =^{\mathrm{mul}} \{t_1, \ldots, t_m\}\restriction_*$ and

Use the defined order recursively (like AC-RPO)

Theorem

$>_{\mathrm{ACKBO}}$ is an AC-simplification order.

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

Correctness of [Steinbach '90]

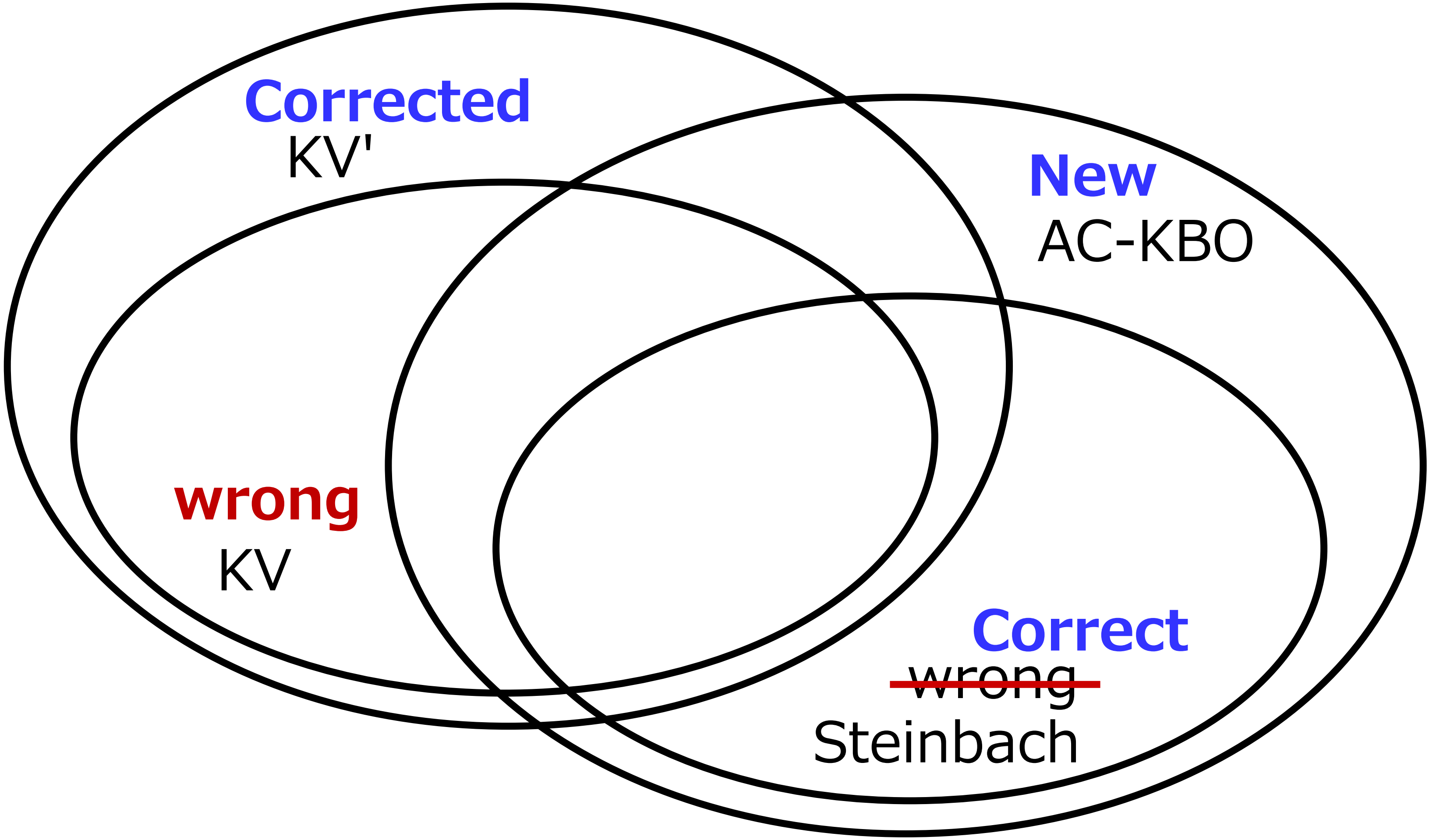
Theorem

If the precondition for Steinbach's theorem holds, then $>_{\mathrm{ACKBO}} = >_{\mathrm{S}}$

Corollary

$>_S$ is a correct AC-simplification order.

Comparison



Complexity results

The orientability problem

Problem (Orientability)

Given $\begin{cases} l_1 \to r_1 \\ \vdots \\ l_n \to r_n \end{cases}$ find an AC-KBO s.t. $\begin{cases} l_1 >_{\mathrm{ACKBO}} r_1 \\ \vdots \\ l_n >_{\mathrm{ACKBO}} r_n \end{cases}$

Note: KBO orientability is P [Korovin & Voronkov '03]

Theorem

AC-KBO orientability is NP-hard.

Proof:

By reduction from a CNF satisfiability problem.

AC-KBO again

$s >_{\mathrm{ACKBO}} t$ 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) $s = f(s_1, \ldots), t = f(t_1, \ldots), (s_1, \ldots) >^{lex}_{\mathrm{ACKBO}} (t_1, \ldots)$,
  - (3) $s = s_1 * \cdots * s_n, t = t_1 * \cdots * t_m$,
    - $\{s_1, \ldots, s_n\} \restriction_* >^{\mathrm{mul}}_{\mathrm{ACKBO}} \{\mathrm{t}_1, \ldots, \mathrm{t}_\mathrm{m}\} \restriction_*$, or
      $\{s_1, \ldots, s_n\} \restriction_* =^{\mathrm{mul}}_{\mathrm{AC}} \{t_1, \ldots, t_m\} \restriction_*$, and
      - $n > m$, or
      - $n = m$ and $\{s_1, \ldots, s_n\} >^{\mathrm{mul}}_{\mathrm{ACKBO}} \{t_1, \ldots, t_m\}$.

The operation $\upharpoonleft_*$

- A high-pass filter w.r.t. precedence

  called BigHead/NoSmallHead in AC-RPO [Rubio '02]

- Example:

  if $f > g > h > i > j$ then

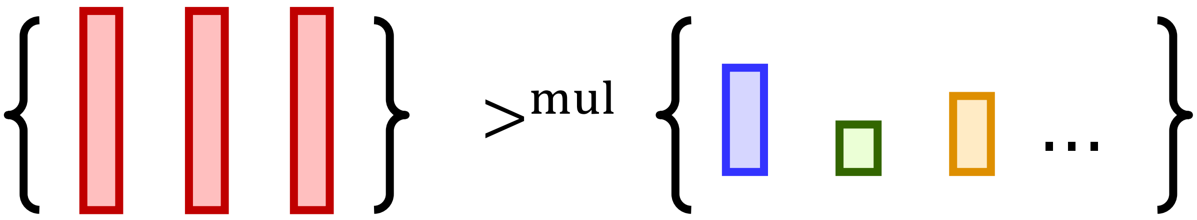
  $$\{f(x), g(x), i(x), j(x)\} \upharpoonleft_h = \{f(x), g(x)\}$$

  Let us assume $\mathrm{True} > * > False$. Then,

  $$\{T, T, F, F, T\} \upharpoonleft_* = \{T, T, T\}$$

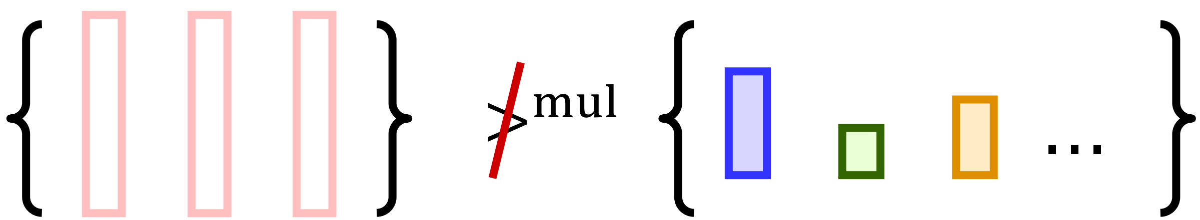
Encoding positive literals

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

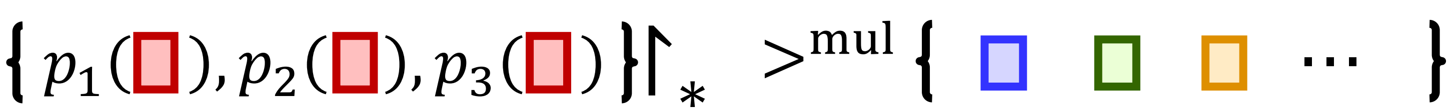


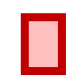
Encoding positive literals

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



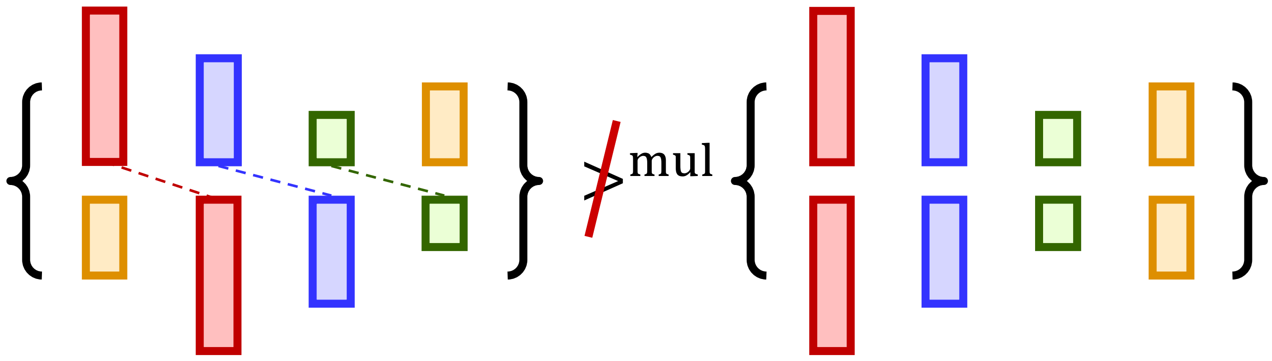
- Thus, $p_1 \lor p_2 \lor p_3$ iff



by choosing big enough 

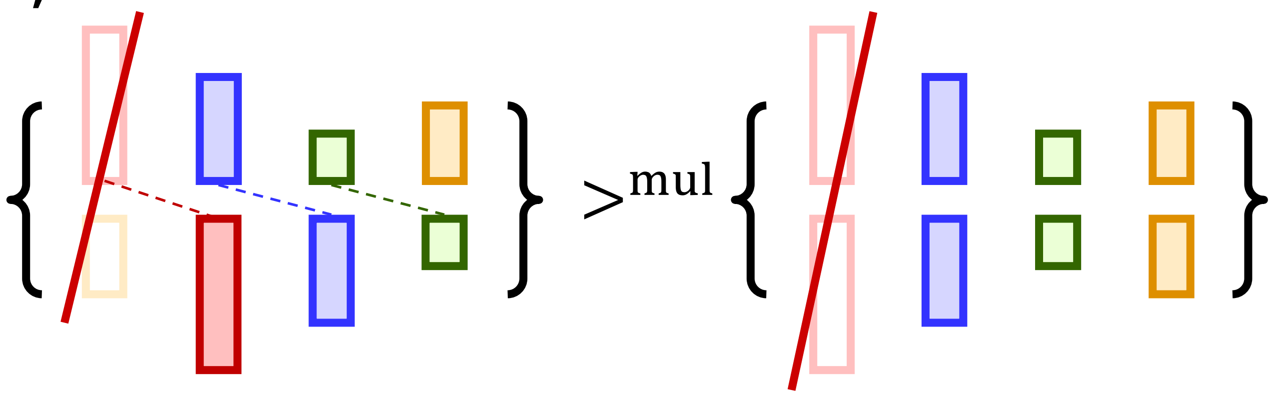
Encoding negative literals

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

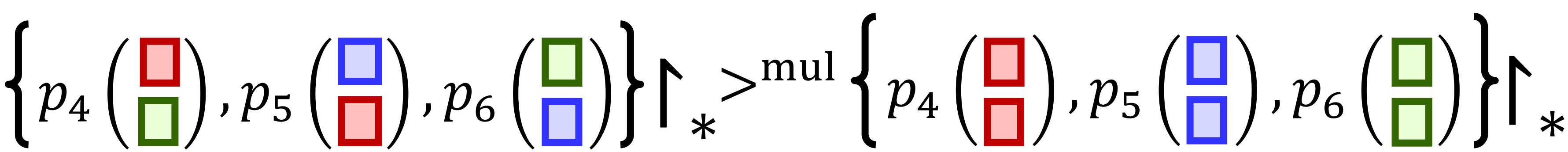


Encoding negative literals

- Consider $p_1 \lor p_2 \lor p_3 \lor \underline{\neg p_4 \lor \neg p_5 \lor \neg p_6}$
- But,



- Thus, $\neg p_4 \lor \neg p_5 \lor \neg p_6$ iff



with choosing big enough ▭, ▭, or ▭

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 $>_{\mathrm{KV}}$
- For $>_{\mathrm{KV}'}$ even testing $s >_{\mathrm{KV}'} t$ is NP-hard

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

	KBO	Steinbach	AC-KBO	KV	KV'
membership	Linear [Löchner '06]	**P**	**P**	**P**	**NP-hard**
orientability	P [KV '03]	?	**NP-hard**	**NP-hard**	**NP-hard**

An extension, implementation, and experiments

Transfinite KBO [Ludwig & Waldmann'07]

- Weight by *linear ordinal polynomial*:
  - $\mathsf{w}(x) = x$,
  - $\mathsf{w}(f(s_1,\ldots, s_n)) =$

$$\mathsf{w}_f \oplus \mathbf{c}_{f,1} \odot \mathsf{w}(s_1) \oplus \ldots \oplus \mathbf{c}_{f,n} \odot \mathsf{w}(s_n)$$

(ordinal) coefficients can be given

- [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**: $s =_{AC} t > u =_{AC} v \Rightarrow s > v$

  - $\mathrm{w}(x * (y * z)) = \mathrm{w}((x * y) * z)$ should hold, i.e.,

$$(1 + c_{*,2})w_* + c_{*,1}x + c_{*,2}\, c_{*,1}\, y + (c_{*,2})^2 z$$
$$= (1 + c_{*,1})w_* + (c_{*,1})^2 x + c_{*,1}\, c_{*,2}\, y + c_{*,2}\, z$$

$\mathbf{c_{*,1} = 1}$ $\mathbf{c_{*,2} = 1}$

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

method	orientability			AC-DP			AC-completion		
	yes	time	T.O.	yes	time	T.O.	yes	time	T.O.
Steinbach	23	**1.6**	0	50	463.2	2	24	2235.4	36
KV	30	2.0	0	66	474.3	4	25	2279.4	37
KV'	30	2.1	0	66	472.4	3	25	2279.6	37
AC-KBO	32	1.7	0	66	**463.1**	3	25	2278.6	37
+coefficient	**37**	47.1	0	**68**	464.7	2	**28**	**1724.7**	26
AC-RPO	63	2.8	0	79	501.5	4	28	1701.6	26

- $>_{\mathrm{S}}$ to $>_{\mathrm{KV}}$ is a big improvement.
- $>_{\mathrm{KV}}$ to $>_{\mathrm{KV}'}$ is merely a bugfix.
- $>_{\mathrm{ACKBO}}$ is slightly more efficient than $>_{\mathrm{KV}'}$
- 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!