Making the Knuth-Bendix algorithm exponentially slower

Reinis Cirpons
University of St Andrews

June 2023

Motivation

$$M = \langle A \mid u_1 = v_1, \dots, u_n = v_n \rangle$$

$u_i, v_i \in A^*$

Word problem:

Given $x, y \in A^*$ decide if $x =_M y$ or not.

Motivation

Example

$$M = \langle a, b, c \mid b^2a^2 = ab,\ a^3 = 1,\ ba = c \rangle$$

Do b^2 and ac represent the same element?

Motivation

Example

$$M = \langle a, b, c \mid b^2a^2 = ab,\ a^3 = 1,\ ba = c \rangle$$

Do b^2 and ac represent the same element?

Yes!

$$b^2 = b^2 1 = b^2a^3 = (b^2a^2)a = aba = a(ba) = ac$$

Motivation

Example

$$M = \langle a, b, c \mid b^2a^2 = ab,\ a^3 = 1,\ ba = c \rangle$$

Do b^2 and ac represent the same element?

Yes!

$$b^2 = b^2 1 = b^2a^3 = (b^2a^2)a = aba = a(ba) = ac$$

What about aba^5bc and $c^3ab^3a^7c$?

Motivation

Example

$$M = \langle a, b, c \mid b^2 a^2 = ab,\ a^3 = 1,\ ba = c \rangle$$

Do b^2 and ac represent the same element?

Yes!

$$b^2 = b^2 1 = b^2 a^3 = (b^2 a^2) a = aba = a(ba) = ac$$

What about aba^5bc and $c^3ab^3a^7c$?

No!

Motivation

Theorem (Markov 1947, Post 1947)

There exists a monoid with undecideable WP.

Theorem (Matiyasevich 1967)

The monoid generated by $\{a, b\}$ with relations

$$\begin{aligned}
a(ab)^2 &= ba^2, \\
a^2b^2 &= ba^2, \\
(ba)^{32}(b^{17}aba)^7(ba)^2b^{96} &= bab^3ab^{15}ab^2ab^{21}(ab)^{31}ab^{26}ab(b^2a)^2 \\
&\quad b^{45}ab^{54}ab^2ab^{18}(ab^2)^2b^3(b^4a)^2b^{14}ab^4 \\
&\quad ab^{18}a(b^2ab)^2bab^{16},
\end{aligned}$$

has undecideable WP.

Motivation

Open problem

Does every 1-relation monoid

$$\langle A \mid u = v \rangle \qquad u, v \in A^*$$

have decideable WP?

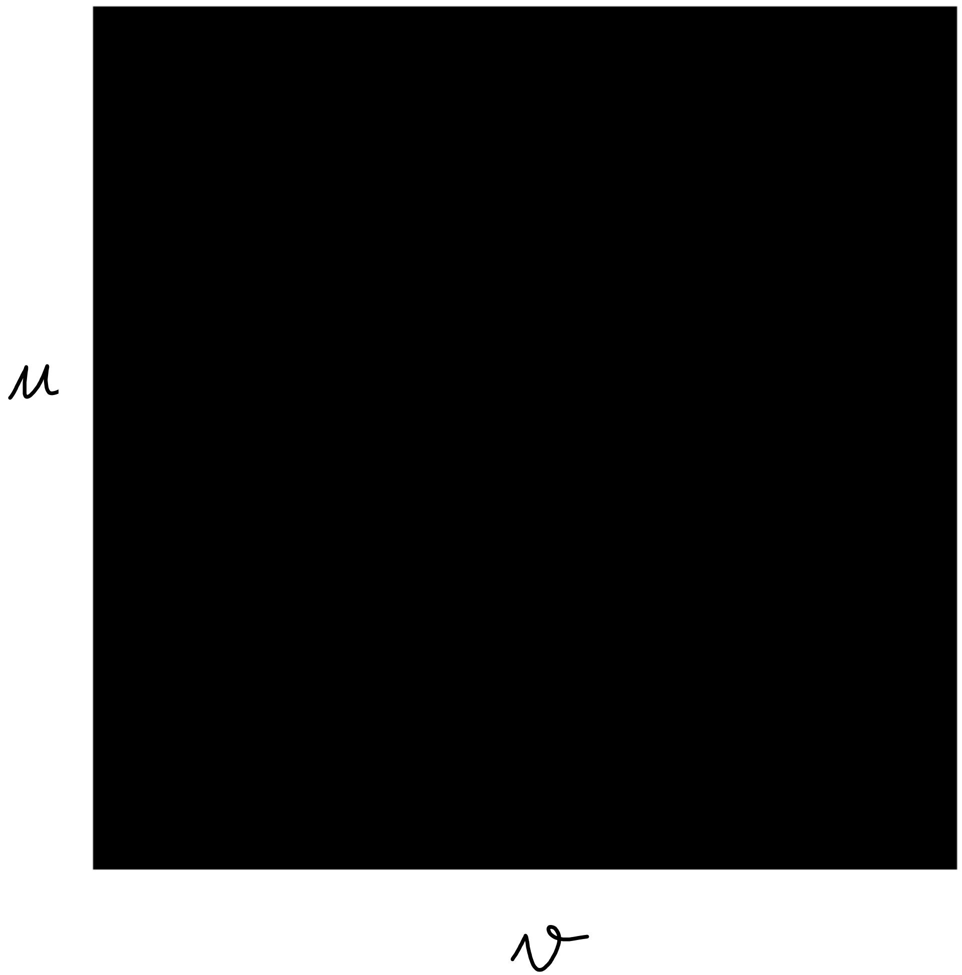
Theorem (Magnus 1932)

The WP is decideable for all 1-relation groups.

For a comprehensive introduction see:

Nyberg-Brodda, Carl-Fredrik (2021), "The word problem for one-relation monoids: a survey", Semigroup Forum, 103 (2): 297–355, arXiv:2105.02853, doi:10.1007/s00233-021-10216-8

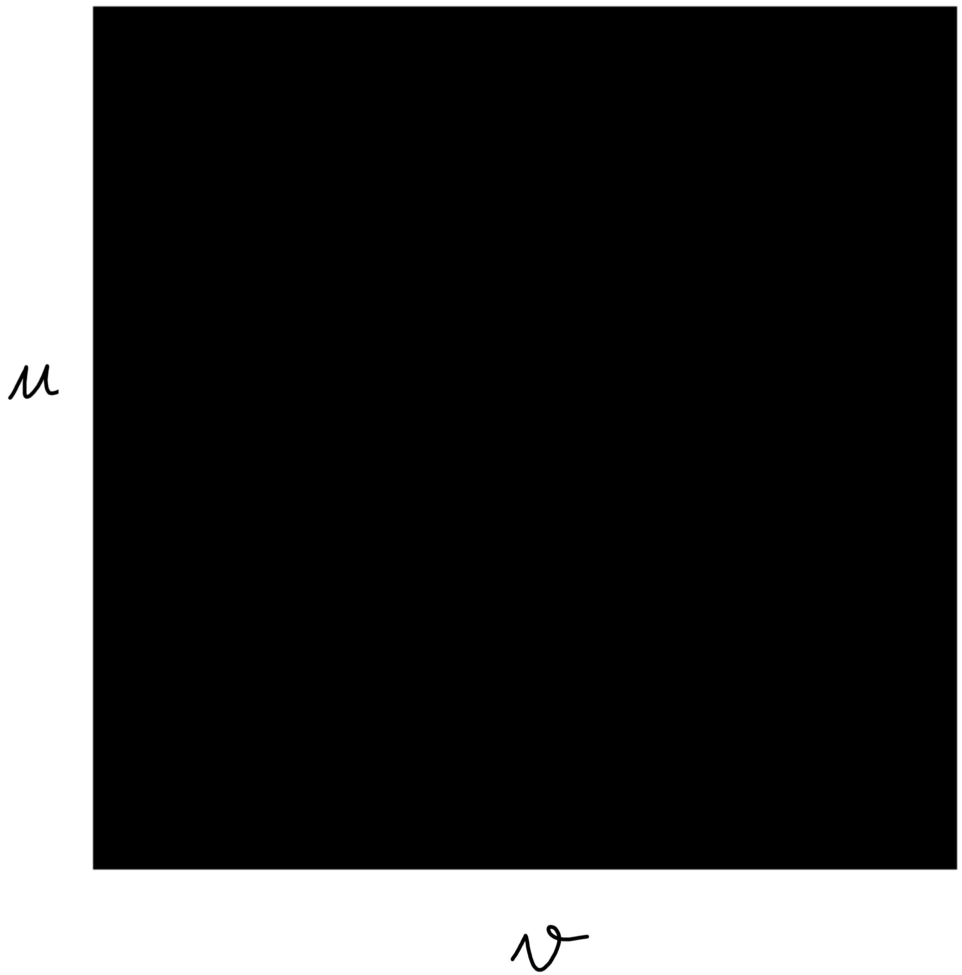
Our efforts



Our efforts

Our efforts

Can we solve the WP in all 2-gen. 1-relation monoids where $|u|, |v| \leq 11$?

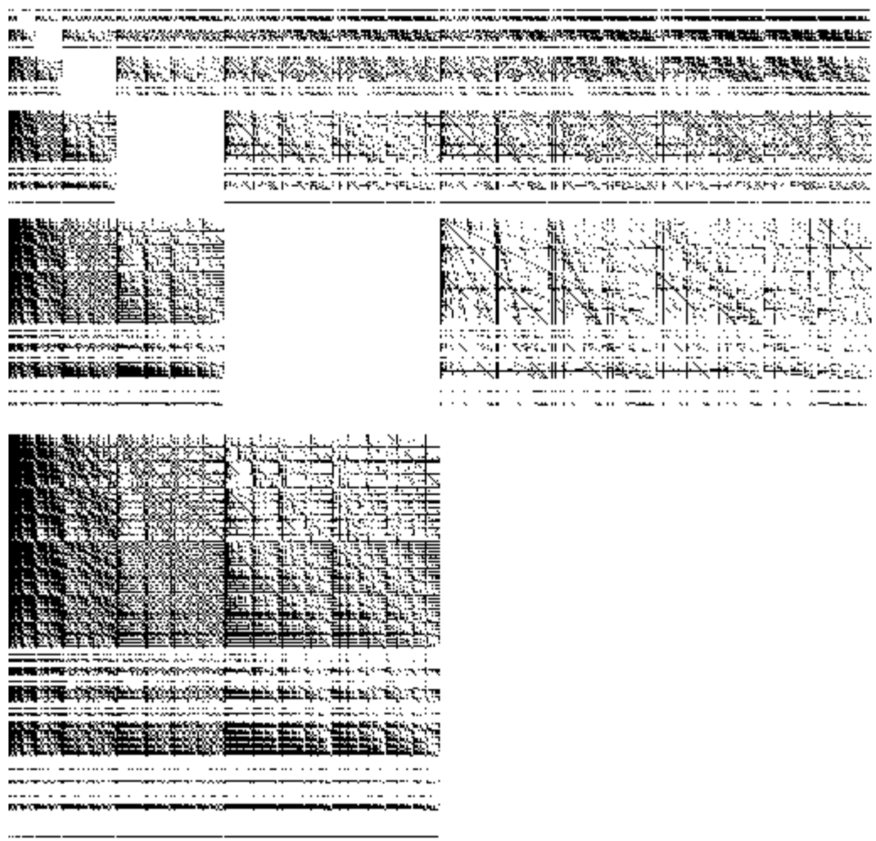


4'190'209 monoids total

Our efforts

All 4'190'209 reduce to one of 42'020.

bna



a, ava

Our efforts

C., James Mitchell, Finn Smith, "The word problem for 1-relation monoids is mostly decidable", WIP

Online encyclopedia of 1-relation monoid presentations

Home | Unsolved Cases | About

Enter a pair of words separated by an "=" using the alphabet $\{a, b\}$, e.g. $baa = aa$:

Search

Displaying search results for "abb=aa".

The presentation $\langle\, a, b \mid abb = aa \,\rangle$ is equivalent to $\langle\, a, b \mid bba = aa \,\rangle$ via the following transformations:

▸ Click here to show the transformation steps

Displaying proofs for $\langle\, a, b \mid bba = aa \,\rangle$ instead!

$$\langle\, a, b \mid bba = aa \,\rangle$$

Unconditional proofs of the decideability of the word problem

There are 5 unconditional proofs for the decideability of the word problem in $\langle\, a, b \mid bba = aa \,\rangle$.

1. This proof proceeds by establishing that the Watier condition holds.

   ▸ Click here to expand the proof

   For more details about this proof, see Proof #121314.

2. This proof proceeds by constructing a complete rewriting system for the presentation.

   ▸ Click here to expand the proof

   For more details about this proof, see Proof #121315.

3. This proof proceeds by constructing a complete rewriting system for the presentation.

   ▾ Click here to expand the proof

   - Reversing the words in the presentation:

     $$\langle\, a, b \mid bba = aa \,\rangle$$

     yields

     $$\langle\, a, b \mid aa = abb \,\rangle.$$

42'020 remain...

42'020 remain...

Can we do better?

Rewriting systems

$$R = \left\{ \begin{array}{l} u_1 \longrightarrow v_1, \\ u_2 \longrightarrow v_2, \\ \quad\vdots \\ u_n \longrightarrow v_n \end{array} \right\} \qquad u_i, v_i \in A^*$$

Rewriting systems

$$R = \left\{ \begin{array}{l} u_1 \to v_1, \\ u_2 \to v_2, \\ \quad\vdots \\ u_n \to v_n \end{array} \right\} \qquad u_i, v_i \in A^*$$

Given $x, y \in A^*$ we write $x \to_R y$ if we can factorize

$$x = s\, u_i\, t \quad , \quad y = s\, v_i\, t$$

for some $s, t \in A^*$, $i \in \mathbb{N}$.

Rewriting systems

$$R = \left\{ \begin{array}{l} u_1 \to v_1 , \\ u_2 \to v_2 , \\ \quad \vdots \\ u_n \to v_n \end{array} \right\} \qquad u_i , v_i \in A^*$$

Given $x, y \in A^*$ we write $x \to_R y$ if we can factorize

$$x = s\, u_i\, t \quad , \quad y = s\, v_i\, t$$

for some $s, t \in A^*$, $i \in \mathbb{N}$.

A word $x \in A^*$ is *irreducible* if it cannot be written as

$$x = s\, u_i\, t \quad \text{for any } s, t \in A^*, \; i \in \mathbb{N}.$$

Rewriting systems

$$R = \left\{ \begin{array}{l} u_1 \to v_1, \\ u_2 \to v_2, \\ \quad \vdots \\ u_n \to v_n \end{array} \right\} \qquad u_i, v_i \in A^*$$

Given $x, y \in A^*$ we write $x \to_R y$ if we can factorize

$$x = s\, u_i\, t \quad , \quad y = s\, v_i\, t$$

for some $s, t \in A^*$, $i \in \mathbb{N}$.

A word $x \in A^*$ is irreducible if it cannot be written as

$$x = s\, u_i\, t \quad \text{for any } s, t \in A^*, \ i \in \mathbb{N}.$$

Write $x \xrightarrow{*}_R y$ if there exists a sequence

$$x = z_1 \to_R z_2 \to_R \cdots \to_R z_m = y$$

Rewriting systems

To every rewriting system R, we associate the monoid M

$$R = \left\{ \begin{array}{l} u_1 \to v_1, \\ u_2 \to v_2, \\ \quad \vdots \\ u_n \to v_n \end{array} \right\} \qquad M = \langle A \mid u_1 = v_1, \ldots, u_n = v_n \rangle$$

Rewriting systems

To every rewriting system R, we associate the monoid M

$$R = \left\{ \begin{array}{l} u_1 \to v_1 , \\ u_2 \to v_2 , \\ \vdots \\ u_n \to v_n \end{array} \right\} \qquad M = \langle A \mid u_1 = v_1, \dots, u_n = v_n \rangle$$

Observations

If $x \xrightarrow[R]{*} y$, then $x =_M y$.

Rewriting systems

To every rewriting system R, we associate the monoid M

$$R = \left\{ \begin{array}{l} u_1 \to v_1, \\ u_2 \to v_2, \\ \quad \vdots \\ u_n \to v_n \end{array} \right\} \qquad M = \langle A \mid u_1 = v_1, \ldots, u_n = v_n \rangle$$

Observations

If $x \xrightarrow[R]{*} y$, then $x =_M y$.

If $x =_M y$, then

$$x = z_1 \xrightarrow[R]{*} z_2 \xleftarrow[R]{*} z_3 \xrightarrow[R]{*} z_4 \xleftarrow[R]{*} \cdots \xrightarrow[R]{*} z_{2m} \xleftarrow[R]{*} z_{2m+1} = y$$

Complete rewriting systems

A rewriting system R is

- Terminating if there are no infinite rewriting sequences

$$z_1 \to_R z_2 \to_R z_3 \to_R \cdots \quad \times$$

Complete rewriting systems

A rewriting system R is

- **Terminating** if there are no infinite rewriting sequences

  $z_1 \to_R z_2 \to_R z_3 \to_R \cdots$ ✗

- **Confluent** if $\forall\, x, y_1, y_2 \in A^*$ s.t. $x \to_R^* y_1$, $x \to_R^* y_2$
  $\exists\, z \in A^*$ s.t. $y_1 \to_R^* z$, $y_2 \to_R^* z$, i.e.

$$\begin{tikzcd}
 & x \arrow[dl, "*"', "R"] \arrow[dr, "*", "R"'] & \\
y_1 \arrow[dr, dashed, "*"', "R"] & & y_2 \arrow[dl, dashed, "*", "R"'] \\
 & z &
\end{tikzcd}$$

Complete rewriting systems

A rewriting system R is

- **Terminating** if there are no infinite rewriting sequences

$$z_1 \to_R z_2 \to_R z_3 \to_R \cdots$$

- **Confluent** if $\forall\, x, y_1, y_2 \in A^*$ s.t. $x \to_R^* y_1$, $x \to_R^* y_2$, $\exists\, z \in A^*$ s.t. $y_1 \to_R^* z$, $y_2 \to_R^* z$, i.e.

$$\begin{tikzcd} & x \arrow[dl, "*" ', "R"] \arrow[dr, "*", "R" '] & \\ y_1 \arrow[dr, dashed, "*" ', "R"] & & y_2 \arrow[dl, dashed, "*", "R" '] \\ & z & \end{tikzcd}$$

- **Complete** if it's both confluent and terminating.

Complete rewriting systems

A rewriting system R is

- **Terminating** if there are no infinite rewriting sequences

$$z_1 \to_R z_2 \to_R z_3 \to_R \cdots$$

- **Confluent** if $\forall x, y_1, y_2 \in A^*$ s.t. $x \to_R^* y_1$, $x \to_R^* y_2$, $\exists z \in A^*$ s.t. $y_1 \to_R^* z$, $y_2 \to_R^* z$, i.e.

$$\begin{tikzcd}
 & x \arrow[dl, "*"', "R"] \arrow[dr, "*", "R"'] & \\
y_1 \arrow[dr, dashed, "*"', "R"] & & y_2 \arrow[dl, dashed, "*", "R"'] \\
 & z &
\end{tikzcd}$$

- **Complete** if it's both confluent and terminating.

Theorem

The WP is decidable in the associated monoid of any complete rws.

Complete rewriting systems

Issue: It is undecideable if a given rws is terminating

Complete rewriting systems

Issue: It is undecideable if a given rws is terminating

— // —

is confluent.

Complete rewriting systems

Issue: It is undecideable if a given rws is terminating

— // — is confluent.

Def: A rws R is **locally confluent** if

$\forall\, x, y_1, y_2 \in A^*$ s.t. $x \to_R y_1$, $x \to_R y_2$

$\exists\, z \in A^*$ s.t. $y_1 \xrightarrow{*}_R z$, $y_2 \xrightarrow{*}_R z$, i.e.

$$\begin{tikzcd}
& x \arrow[dl, "R"'] \arrow[dr, "R"] & \\
y_1 \arrow[dr, dashed, "*"', "R"] & & y_2 \arrow[dl, dashed, "*", "R"'] \\
& z &
\end{tikzcd}$$

Complete rewriting systems

Issue: It is undecideable if a given rws is terminating
—//— is confluent.

Def: A rws R is **locally confluent** if
$\forall x, y_1, y_2 \in A^*$ s.t. $x \to_R y_1$, $x \to_R y_2$
$\exists z \in A^*$ s.t. $y_1 \xrightarrow{*}_R z$, $y_2 \xrightarrow{*}_R z$, i.e.

$$\begin{tikzcd} & x \arrow[dl, "R"'] \arrow[dr, "R"] & \\ y_1 \arrow[dr, dashed, "*"', "R"] & & y_2 \arrow[dl, dashed, "*", "R"'] \\ & z & \end{tikzcd}$$

Lemma (Newman 1942)
Locally confluent and terminating $\Rightarrow$ confluent.

Observation
Local confluence is decideable if R is terminating.

Critical pairs

Example

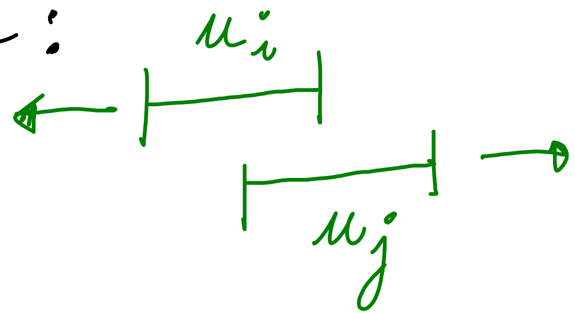
$$R = \{\, bbaa \to ab,\ aaa \to 1,\ ba \to c \,\}$$

Critical pairs

Example

$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$

Idea:



Critical pairs

Example

$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$

Idea:

$$\leftarrow bbaa$$

$$aaa \to$$

Critical pairs

Example

$$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$$

Idea:

$$\begin{array}{l} \quad\ \leftarrow bbaa \\ aaa \rightarrow \end{array}$$

Critical pairs

Example

$$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$$

Idea:

$$\leftarrow \ bbaa$$

$$aaaa \ \to \quad X$$

Critical pairs

Example

$$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$$

Idea:

$$\leftarrow bbaa$$
$$aaa \rightarrow$$

X

Critical pairs

Example

$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$

Idea:

$\leftarrow bbaa$

$aaa \to$ X

Critical pairs

Example

$$R = \{\, bbaa \to ab,\ aaa \to 1,\ ba \to c \,\}$$

Idea:

$$\begin{array}{l} \leftarrow bb\boxed{aa} \\ \boxed{aa}a \rightarrow \end{array} \qquad \checkmark$$

$$\begin{array}{c} bbaaa \\ \swarrow \qquad \searrow \end{array}$$

Critical pairs

Example

$R = \{ bbaa \to ab, \; aaa \to 1, \; ba \to c \}$

Idea: $\leftarrow bba\boxed{a}$

$\boxed{a}aa \to$ ✓

$bbaaa$ $\qquad$ $bbaaaa$

Critical pairs

Example

$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$

Idea:

$bbaaa$ $bbaaaa$ $bbaa$ $aaaa$

$aaaaa$ $baaa$

Critical pairs

Example

$$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$$

Idea:

$$aba \leftarrow bbaaa \rightarrow bb$$

$$abaa \leftarrow bbaaaaa \rightarrow bba$$

$$ab \leftarrow bbaa \rightarrow bca$$

$$a \leftarrow aaaa \rightarrow a$$

$$aa \leftarrow aaaaa \rightarrow aa$$

$$caa \leftarrow baaa \rightarrow b$$

Critical pairs

Example

$$R = \{\, bbaa \to ab,\ aaa \to 1,\ ba \to c \,\}$$

Idea:

$$\begin{tikzcd}[column sep=small] & bbaaa \arrow[dl] \arrow[dr] & \\ aba \arrow[d] & & bb \\ ac & & \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & bbaaaaa \arrow[dl] \arrow[dr] & \\ abaa \arrow[d] & & bba \\ aca & & \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & bbaa \arrow[dl] \arrow[dr] & \\ ab & & bca \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & aaaa \arrow[dl] \arrow[dr] & \\ a & = & a \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & aaaaa \arrow[dl] \arrow[dr] & \\ aa & = & aa \end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small] & baaa \arrow[dl] \arrow[dr] & \\ caa & & b \end{tikzcd}$$

Not locally confluent!

Critical pairs

$$\begin{tikzcd}
 & bbaaa \arrow[dl] \arrow[dr] & \\
aba \arrow[d] & & bb \\
ac & &
\end{tikzcd}
\qquad \Rightarrow \qquad ac =_M bb$$

Critical pairs

$$
\begin{tikzcd}
& bbaaa \arrow[dl] \arrow[dr] & \\
aba \arrow[d] & & bb \\
ac & &
\end{tikzcd}
$$

$\Rightarrow \quad ac =_M bb$

So can add

$ac \to bb$

or

$bb \to ac$

without changing underlying monoid.

Critical pairs

$$\begin{tikzcd}
 & bbaaa \arrow[dl] \arrow[dr] & \\
aba \arrow[d] & & bb \arrow[d] \\
ac & = & ac
\end{tikzcd}$$

$\Rightarrow \quad ac =_M bb$

So can add

$ac \to bb$

or

$bb \to ac$

without changing underlying monoid.

Let's add $bb \to ac$

Knuth-Bendix algorithm

Input: terminating rws R

1. While R is not locally confluent:
2. Pick an unresolved critical pair

$$x \xleftarrow[R]{*} c \xrightarrow[R]{*} y, \quad x \neq y$$

3. Add either $x \to y$ or $y \to x$ to R,
   preserving termination
4. Return R

How to preserve termination?

Option 1: Termination orders

Example: Len - lex ordering

Assign an order to alphabet, say $a < b < c$

Define $>_{lenlex}$ on A^* by

$x >_{lenlex} y$ if $\mathrm{len}(x) > \mathrm{len}(y)$

or if $\mathrm{len}(x) = \mathrm{len}(y)$ and y comes before x in a dictionary.

E.g. $aab >_{lenlex} ba$

$bca >_{lenlex} acc$

How to preserve termination?

Importantly $x >_{\text{lenlex}} y \Rightarrow sxt >_{\text{lenlex}} syt$

and every infinite set contains its minimum.

How to preserve termination?

Importantly $x >_{\text{lenlex}} y \Rightarrow s x t >_{\text{lenlex}} s y t$ and every infinite set contains its minimum.

Other examples: Weights, RPO etc.

How to preserve termination?

Importantly $x >_{lenlex} y \Rightarrow sxt >_{lenlex} syt$ and every infinite set contains its minimum.

Other examples: weights, RPO etc.

Observation:

If a rws R preserves a term. order $>$, then it is terminating.

How to preserve termination?

Importantly $x >_{lenlex} y \Rightarrow sxt >_{lenlex} syt$ and every infinite set contains its minimum.

Other examples: Weights, RPO etc.

Observation:

If a rws R preserves a term. order $>$, then it is terminating.

When adding $x =_M y$ to R, pick $x \to y$ if $x > y$ and $y \to x$ if $y > x$.

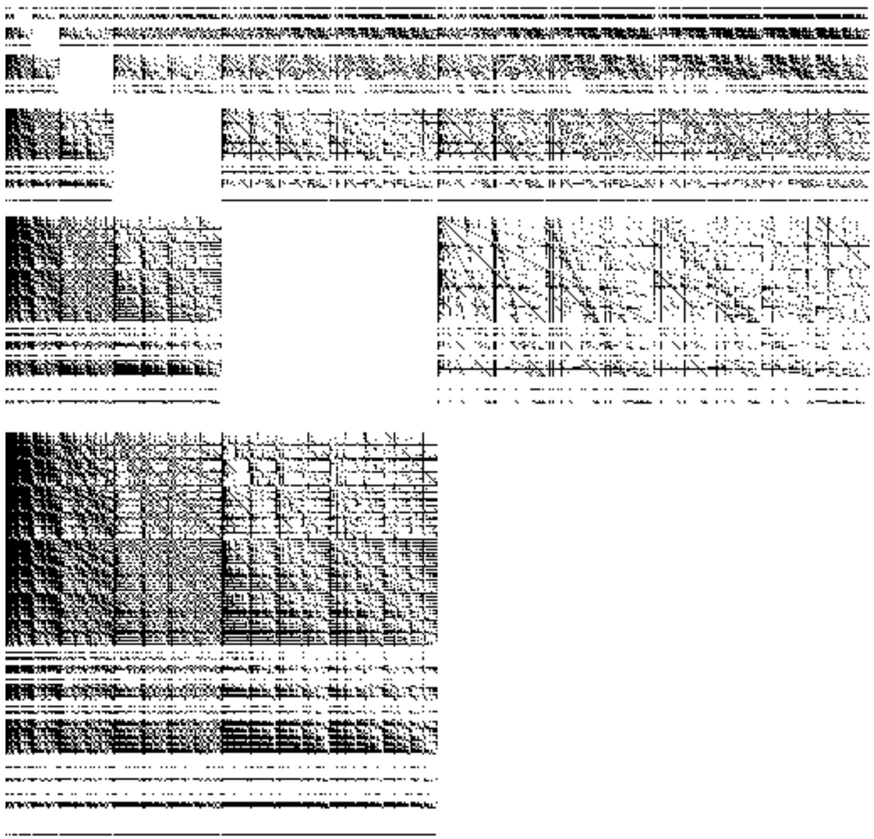
How to reserve termination?

$$R = \{ bbaa \to ab,\ aaa \to 1,\ ba \to c \}$$

$$\Big\downarrow \text{KB}$$

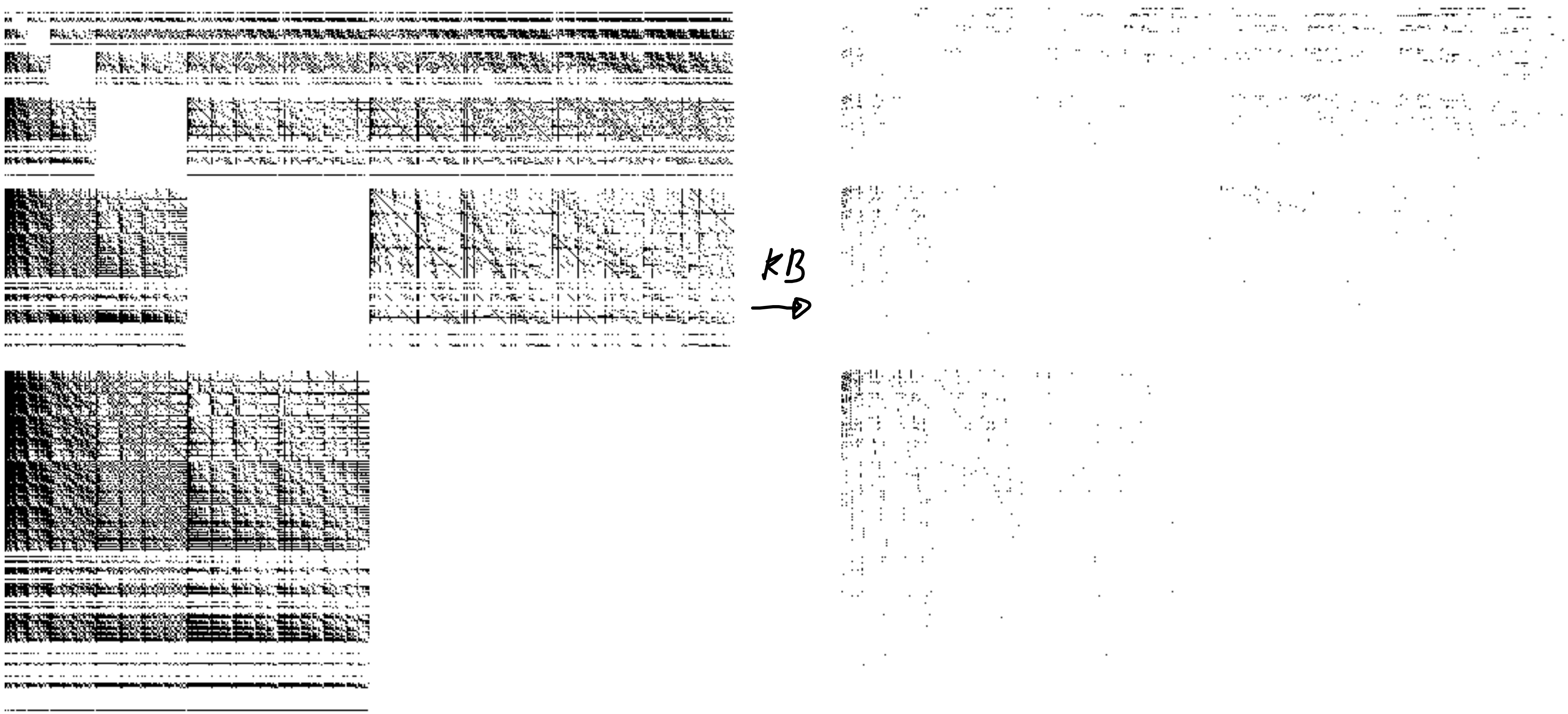
$$R' = \{ ba \to c,\ bb \to ac,\ aaa \to 1,\ aca \to bc,$$
$$acb \to cc,\ bca \to ab,\ caa \to b,$$
$$cca \to acc,\ ccb \to bcc,\ aabc \to ca,$$
$$aacc \to cb,\ bcbc \to aab,\ bccc \to abcb,$$
$$cabc \to ab,\ cacc \to bcb,\ acccc \to cbcb,$$
$$ccccc \to c \}$$

Knuth-Bendix with termination ordering



4.2.020

Knuth-Bendix with termination ordering



42'020

1'226

Knuth - Bendix with termination ordering

Smallest one we can't do with KB:

$\langle a, b \mid baa\ baa = aba \rangle$

Can we do better?

Termination solvers

Termination Competition 2022 [Show configs] [Show scores] [One column]

Competition-Wide Ranking

AProVE+LoAT(4.0811) MU-TERM(1.9331) TTT2+TcT(1.9082) NaTT(1.4268) Matchbox(1.3425) iRankFinder(1.2594) Ultimate(1.2079) MultumNonMulta(1.1930) NTI+cTI(0.9649) SOL(0.9180) Wanda(0.8975)

Advancing-the-State-of-the-Art Ranking

Matchbox(67) MultumNonMulta(48) AProVE+LoAT(31.25) SOL(16) NaTT(1) NTI+cTI(1) TTT2+TcT(0.375) iRankFinder(0) MU-TERM(0) Ultimate(0) Wanda(0)

Termination of Rewriting Progress: 100%, CPU Time: 85d 8:05:33, Node Time: 34d 3:49:50

TRS Standard 54200 54199
1. AProVE21
✔1. AProVE21
2. NaTT 2.3.2
3. ttt2-1.20
✔2. ttt2-1.20
4. muterm 6.0.3
✔3. NaTT 1.6.2
5. NTI_22

SRS Standard 54202 54201
1. matchbox-2022-07-22
✔1. matchbox-2022-07-22
2. MnM3.19c
3. AProVE21
✔2. AProVE21
4. ttt2-1.20
✔3. ttt2-1.20
5. NaTT 2.3.2
✔4. NaTT 1.6.2
6. muterm 6.0.3

TRS Relative 54195 54196
1. NaTT 2.3.2
2. AProVE21
✔1. AProVE21
✔2. ttt2-1.20
3. ttt2-1.20

SRS Relative 54198 54197
1. MnM3.19c
2. AProVE21
✔1. AProVE21
3. ttt2-1.20
✔2. ttt2-1.20
4. NaTT 2.3.2

TRS Equational 54203 54204
1. AProVE21
✔1. AProVE21
2. muterm 5.18
3. NaTT 2.3.2

TRS Conditional - Operational Termination 54205
1. MU-TERM 6.1
2. AProVE21

TRS Context Sensitive 54206
1. muterm 5.18
2. AProVE21

TRS Innermost 54208 54207
1. AProVE21
✔1. AProVE21
2. muterm 6.0.3

TRS Outermost 54209 54210
1. AProVE21
✔1. AProVE21

HRS Union Beta 54227
1. SOL 2022
2. Wanda 2.2a

Termination of Programs Progress: 100%, CPU Time: 3d 3:22:33, Node Time: 2d 4:20:44

C 54224
1. Aprove22-C
2. UltimateAutomizer2022v2

C Integer 54225
1. Aprove22-C
2. UltimateAutomizer2022v2
3. irankfinder v1.3.2

Integer Transition Systems 54213
1. irankfinder v1.3.2
2. LoAT TermComp 2021

Logic Programming 54212
1. NTI+cTI_22
2. AProVE21

Complexity Analysis Progress: 100%, CPU Time: 129d 22:10:39, Node Time: 42d 19:13:03

Derivational Complexity: TRS 54215 54214
1. AProVE21
✔1. tct-trs_v3.2.0_2020-06-28
2. tct-trs_v3.2.0_2020-06-28

Derivational Complexity: TRS Innermost 54221 54217
1. AProVE21
✔1. tct-trs_v3.2.0_2020-06-28
2. tct-trs_v3.2.0_2020-06-28

Runtime Complexity: TRS 54218 54216
1. AProVE21
2. tct-trs_v3.2.0_2020-06-28
✔1. tct-trs_v3.2.0_2020-06-28

Runtime Complexity: TRS Innermost 54226 54220
1. AProVE21
2. tct-trs_v3.2.0_2020-06-28
✔1. tct-trs_v3.2.0_2020-06-28
✔2. AProVE21

Demonstrations Progress: 100%, CPU Time: 6d 17:00:21, Node Time: 2d 16:08:07

TRS Conditional - Termination 54222
1. MU-TERM 6.1

Integer TRS Innermost 54233
1. AProVE21

Haskell 54237
1. AProVE21

Java Bytecode 54234
1. AProVE21

Java Bytecode Recursive 54235
1. AProVE21

Logic Programming with Cut 54236
1. AProVE21

Prolog 54238
1. AProVE21

Complexity: C Integer 54241
1. Aprove22-CFR-fix

Complexity: ITS 54242
1. Aprove22-CFR-fix

Runtime Complexity: TRS Parallel Innermost 54223
1. AProVE 2022 parallel-innermost runtime complexity

Termination solvers

TermCOMP 2022: SRS Standard

54202 [Job info CSV] 54201 [Job info CSV] Showing all results.

benchmark	VBS	muterm 6.0.3 default	NaTT 2.3.2 default	ttt2-1.20 ttt2	matchbox-2022-07-22 std.sh	AProVE21 standard	MnM3.19c default	ttt2-1.20 ttt2_cert	AProVE21 certified	NaTT 1.6.2 Certifiable	matchbox-2022-07-22 std-cert.sh	~Y2021
Secret_05_SRS/aprove1.xml	YES	timeout (wallclock)	MAYBE 0.49/0.41	YES 179.68/45.43	YES 0.81/0.47	YES 20.29/5.97	YES 4.46/1.38	YES 30.45/7.90 ✔0.01	YES 20.35/5.89 ✔0.02	MAYBE 0.26/0.22	YES 4.74/2.64 ✔0.03	YES
Secret_05_SRS/aprove2.xml	YES	timeout (wallclock)	MAYBE 0.75/0.67	YES 12.91/3.52	YES 0.18/0.12	YES 23.02/13.14	YES 4.86/1.47	YES 32.29/8.36 ✔0.02	YES 19.10/5.60 ✔0.01	MAYBE 0.27/0.23	YES 0.88/0.52 ✔0.01	YES
Secret_05_SRS/aprove3.xml	YES	timeout (wallclock)	MAYBE 1.03/0.92	YES 25.88/6.78	YES 7.98/4.05	YES 17.52/5.23	YES 12.46/3.40	YES 25.91/6.73 ✔0.02	YES 16.27/4.90 ✔0.01	MAYBE 0.26/0.24	YES 8.95/4.53 ✔0.03	YES
Secret_05_SRS/aprove4.xml	YES	timeout (wallclock)	YES 0.32/0.28	MAYBE 196.31/49.54	YES 9.44/5.26	YES 27.91/7.84	YES 17.58/5.63	YES 39.81/10.28 ✔0.01	YES 24.04/6.83 ✔0.02	MAYBE 0.37/0.31	YES 11.48/6.83 ✔0.02	YES
Secret_05_SRS/aprove5.xml	YES	YES 145.95/146.99	YES 0.44/0.44	YES 185.37/46.98	YES 15.16/7.40	YES 26.68/7.55	YES 9.88/2.79	YES 34.49/8.91 ✔0.02	YES 27.85/7.80 ✔0.03	MAYBE 0.47/0.43	YES 16.70/7.87 ✔0.03	YES
Secret_05_SRS/jambox1.xml	YES	timeout (wallclock)	MAYBE 0.98/0.93	MAYBE 198.10/49.84	YES 7.09/3.53	YES 29.10/8.10	timeout (wallclock)	MAYBE 951.36/300.30	YES 25.93/7.36 ✔0.02	MAYBE 0.37/0.34	YES 7.82/3.83 ✔0.00	YES
Secret_05_SRS/jambox2.xml	YES	timeout (wallclock)	MAYBE 4.02/3.81	MAYBE 198.36/50.05	YES 4.13/2.43	timeout (wallclock)	timeout (wallclock)	MAYBE 952.57/300.30	MAYBE 1163.26/292.50	MAYBE 0.84/0.76	timeout (wallclock)	YES
Secret_05_SRS/jambox3.xml	YES	timeout (wallclock)	MAYBE 1.07/0.96	YES 41.65/10.69	YES 1.31/0.75	YES 92.04/23.88	timeout (wallclock)	MAYBE 953.07/300.30	YES 87.86/22.86 ✔0.01	MAYBE 0.48/0.42	YES 64.72/25.19 ✔0.01	YES
Secret_05_SRS/jambox4.xml	YES	timeout (wallclock)	MAYBE 0.52/0.47	YES 4.25/1.33	YES 0.78/0.44	YES 5.71/2.12	YES 18.82/4.98	YES 28.95/7.50 ✔0.01	YES 36.45/14.34 ✔0.01	MAYBE 0.26/0.23	YES 1.41/0.83 ✔0.01	YES
Secret_05_SRS/jambox5.xml	YES	timeout (wallclock)	MAYBE 0.72/0.61	YES 12.27/3.34	YES 0.59/0.35	YES 43.65/11.71	timeout (wallclock)	MAYBE 952.85/300.31	YES 42.67/11.62 ✔0.01	MAYBE 0.34/0.28	YES 21.69/11.36 ✔0.01	YES
Secret_05_SRS/matchbox1.xml	YES	YES 12.45/12.93	YES 0.21/0.21	YES 7.96/2.35	YES 4.90/2.62	YES 26.27/7.40	YES 7.12/2.06	YES 9.68/2.71 ✔0.00	YES 27.33/7.65 ✔0.01	YES 2.42/1.91 ✔0.02	YES 3.64/2.04 ✔0.02	YES
Secret_05_SRS/matchbox2.xml	YES	YES 4.02/4.22	YES 0.51/0.44	YES 5.50/1.64	YES 0.09/0.08	YES 6.52/2.40	YES 2.42/0.86	YES 7.33/2.07 ✔0.00	YES 22.84/6.55 ✔0.01	YES 1.57/1.45 ✔0.03	YES 11.45/6.69 ✔0.04	YES
Secret_05_SRS/torpa1.xml	YES	YES 2.20/2.37	MAYBE 0.93/0.86	YES 3.94/1.31	YES 0.34/0.21	YES 6.32/2.29	YES 7.87/2.21	YES 21.28/5.58 ✔0.02	YES 19.55/5.73 ✔0.00	MAYBE 0.41/0.38	YES 9.57/5.04 ✔0.03	YES
Secret_05_SRS/torpa2.xml	YES	timeout (wallclock)	MAYBE 0.33/0.31	MAYBE 197.74/49.78	YES 2.62/2.01	YES 18.04/5.32	YES 9.44/2.67	YES 48.92/12.49 ✔0.00	YES 20.62/5.93 ✔0.00	MAYBE 0.14/0.13	YES 2.29/1.33 ✔0.00	YES
Secret_05_SRS/torpa3.xml	YES	YES 4.24/4.60	YES 0.07/0.30	YES 7.22/2.09	YES 0.68/0.40	YES 13.45/4.16	YES 7.62/2.17	YES 13.64/3.66 ✔0.02	YES 13.87/4.27 ✔0.00	YES 0.04/0.04 ✔0.00	YES 0.76/0.46 ✔0.11	YES
Secret_05_SRS/torpa4.xml	YES	timeout (wallclock)	YES 0.19/0.20	YES 9.54/2.67	YES 0.08/0.07	YES 17.94/5.42	YES 6.23/1.80	YES 21.37/5.68 ✔0.01	YES 21.92/6.31 ✔0.00	MAYBE 0.59/0.54	YES 0.09/0.08 ✔0.00	YES
Zantema_04/syracuse.xml	MAYBE	timeout (wallclock)	MAYBE 1.26/1.21	MAYBE 198.67/49.99	timeout (wallclock)	timeout (wallclock)	timeout (wallclock)	MAYBE 953.02/300.30	MAYBE 1170.62/293.93	MAYBE 0.67/0.64	timeout (wallclock)	MAYBE
Zantema_04/z001.xml	YES	timeout (wallclock)	MAYBE 0.73/0.65	YES 1.08/0.52	YES 0.04/0.05	YES 5.26/2.03	YES 2.71/0.94	YES 23.18/6.05 ✔0.26	YES 79.28/20.78 ✔0.04	MAYBE 0.33/0.29	YES 9.47/4.43 ✔0.00	YES
Zantema_04/z002.xml	YES	YES 15.45/15.74	YES 0.05/0.05	YES 0.90/0.50	YES 0.17/0.13	YES 5.65/2.12	YES 0.81/0.43	YES 4.93/1.49 ✔0.01	YES 6.09/2.35 ✔0.00	YES 0.25/0.22 ✔0.00	YES 1.56/0.93 ✔0.01	YES
Zantema_04/z003.xml	YES	timeout (wallclock)	YES 0.08/0.09	YES 2.42/1.12	YES 0.22/0.15	YES 5.60/2.08	YES 1.67/0.66	YES 9.14/2.53 ✔0.00	YES 13.78/4.21 ✔0.00	MAYBE 0.31/0.29	YES 1.17/0.68 ✔0.01	YES
Zantema_04/z004.xml	YES	YES 0.07/0.09	YES 0.14/0.13	YES 0.79/0.46	YES 0.16/0.10	YES 5.48/2.08	YES 1.21/0.54	YES 8.46/2.36 ✔0.00	YES 15.70/4.74 ✔0.00	YES 0.18/0.17 ✔0.00	YES 3.15/1.80 ✔0.00	YES
Zantema_04/z005.xml	YES	YES 59.32/60.30	YES 0.45/0.39	YES 1.48/0.63	YES 0.21/0.14	YES 5.73/2.19	YES 0.91/0.44	YES 21.39/5.60 ✔0.00	YES 13.43/4.20 ✔0.03	YES 0.27/0.23 ✔0.00	YES 1.34/0.78 ✔0.00	YES
Zantema_04/z006.xml	YES	YES 0.03/0.04	YES 0.04/0.04	YES 0.99/0.50	YES 0.06/0.06	YES 5.43/2.18	YES 0.72/0.39	YES 1.99/0.73 ✔0.00	YES 9.80/3.22 ✔0.00	YES 0.02/0.02 ✔0.00	YES 0.05/0.06 ✔0.00	YES
Zantema_04/z007.xml	YES	YES 0.01/0.02	YES 0.04/0.04	YES 1.25/0.57	YES 0.05/0.06	YES 5.33/2.04	YES 0.81/0.43	YES 2.74/0.92 ✔0.00	YES 9.43/3.14 ✔0.00	YES 0.02/0.02 ✔0.00	YES 0.05/0.05 ✔0.00	YES
Zantema_04/z008.xml	YES	YES 135.83/63.21	YES 1.15/1.11	YES 4.16/1.32	YES 0.84/0.50	YES 16.50/4.97	YES 3.02/0.99	YES 15.03/4.01 ✔0.00	YES 20.11/5.85 ✔0.01	YES 0.68/0.64 ✔0.01	YES 2.14/1.20 ✔0.01	YES
Zantema_04/z009.xml	YES	YES 0.04/0.46	YES 0.05/0.05	YES 2.18/0.81	YES 0.24/0.15	YES 11.99/3.88	YES 1.44/0.60	YES 3.06/1.36 ✔0.00	YES 16.04/4.92 ✔0.00	YES 0.03/0.03 ✔0.00	YES 0.97/0.56 ✔0.00	YES
Zantema_04/z010.xml	YES	YES 0.03/0.04	YES 0.04/0.04	YES 1.11/0.54	YES 0.05/0.05	YES 11.88/3.82	YES 0.85/0.46	YES 2.90/0.96 ✔0.00	YES 11.77/3.78 ✔0.00	YES 0.02/0.02 ✔0.00	YES 0.06/0.06 ✔0.00	YES
Zantema_04/z011.xml	YES	YES 0.14/0.14	YES 0.07/0.07	YES 1.95/0.74	YES 0.20/1.29	YES 14.94/4.53	YES 1.68/0.66	YES 4.83/1.44 ✔0.00	YES 15.05/4.55 ✔0.01	YES 0.04/0.04 ✔0.00	YES 1.17/0.71 ✔0.01	YES
Zantema_04/z012.xml	YES	YES 0.08/0.08	YES 0.05/0.05	YES 0.89/0.49	YES 0.06/0.06	YES 21.33/6.13	YES 2.11/0.80	YES 1.17/0.55 ✔0.00	YES 15.72/4.74 ✔0.00	YES 0.02/0.02 ✔0.00	YES 0.06/0.06 ✔0.00	YES
Zantema_04/z013.xml	YES	YES 0.94/0.94	YES 0.05/0.05	YES 1.73/0.70	YES 0.19/0.13	YES 5.92/2.18	YES 1.37/0.65	YES 8.27/2.31 ✔0.01	YES 15.99/4.79 ✔0.02	YES 0.12/0.11 ✔0.01	YES 0.58/0.36 ✔0.00	YES

Termination solvers

Examples:

- Waldmann, J. (2004). Matchbox: A Tool for Match-Bounded String Rewriting. In: van Oostrom, V. (eds) Rewriting Techniques and Applications. RTA 2004. Lecture Notes in Computer Science, vol 3091. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-25979-4_6

- Giesl, J. et al. (2014). Proving Termination of Programs Automatically with AProVE . In: Demri, S., Kapur, D., Weidenbach, C. (eds) Automated Reasoning. IJCAR 2014. Lecture Notes in Computer Science(), vol 8562. Springer, Cham. https://doi.org/10.1007/978-3-319-08587-6_13

Knuth-Bendix algorithm

Input: terminating rws R

1. While R is not locally confluent:
2. Pick an unresolved critical pair
$$\begin{tikzcd} & c \arrow[dl, "*"', "R"] \arrow[dr, "*", "R"'] & \\ x & & y \end{tikzcd}$$
$$x \neq y$$
3. Add either $x \to y$ or $y \to x$ to R, preserving termination

Explore both options

4. Return R

Example

$$\langle a, b \mid baa\,baa = aba \rangle$$

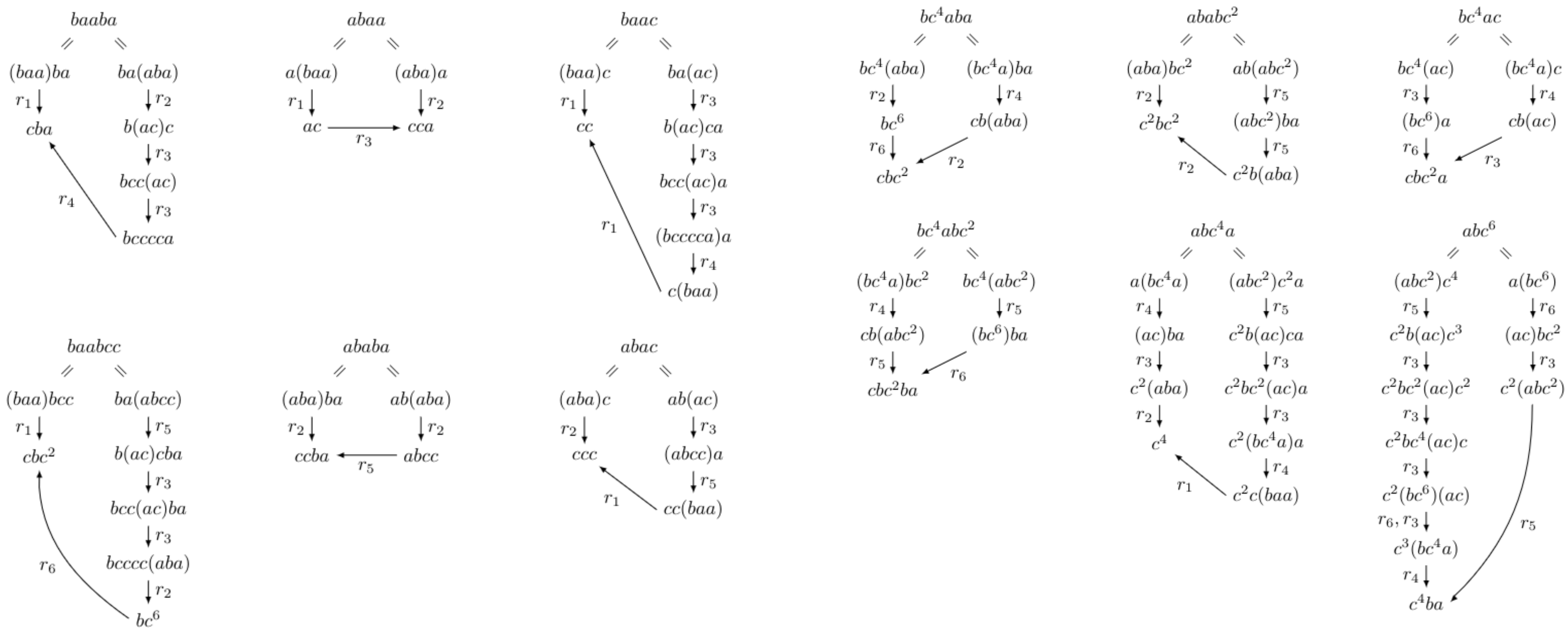
$$\rightsquigarrow$$

$$\langle a, b \mid baa = c \,,\; aba = cc \rangle$$

$$\rightsquigarrow \text{ KB Backtrack}$$

$$\{ ba^2 \to c,\; aba \to c^2,\; ac \to c^2 a,\; bc^4 a \to cba,\; abc^2 \to c^2 ba,\; bc^6 \to cbc^2 \}$$

$$\{\, ba^2 \to c,\ aba \to c^2,\ ac \to c^2a,\ bc^4a \to cba,\ abc^2 \to c^2ba,\ bc^6 \to cbc^2 \,\}$$



With KB-Backtrack can further reduce unsolved presentations to

1'226 $\longrightarrow$ 1'043

This is not a new idea, see e.g.

- Wehrman, I., Stump, A., Westbrook, E. (2006).
  Slothrop: Knuth-Bendix Completion with a Modern Termination Checker.
  In: Pfenning, F. (eds) Term Rewriting and Applications. RTA 2006.
  Lecture Notes in Computer Science, vol 4098. Springer, Berlin, Heidelberg.
  https://doi.org/10.1007/11805618_22

- H. Sato, S. Winkler, M. Kurihara, and A. Middeldorp.
  Constraint-based multi-completion procedures for term rewriting systems.
  IEICE Transactions on Electronics, Information and Communication Engineers,
  E92-D(2):220-234, 2009.

Thank you!