Automated Reasoning for Experimental Mathematics
Part III: AR for the Andrews-Curtis Conjecture

Alexei Lisitsa[1]

[1] Department of Computer Science, The University of Liverpool

28 June 2019

Outline

- Part I: Automated Reasoning for Knots (computational topology)
- Part II: Solution for the Erdős Discrepancy Problem, C=2 (combinatorial number theory)
- **Part III: Exploration of the Andrews-Curtis Conjecture (computational combinatorial group theory)**

Part II is based on

(L18) Lisitsa: The Andrews-Curtis Conjecture, Term Rewriting and First-Order Proofs. ICMS 2018: 343-351

(L19) A. Lisitsa Automated Reasoning for the Andrews-Curtis Conjecture. AITP 2019: 3pp

Groups and their presentations

- Groups are algebraic structures which satisfy the following axioms
  - $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
  - $x \cdot e = x$
  - $e \cdot x = x$
  - $x \cdot x' = e$

- Groups are algebraic structures which satisfy the following axioms
  - $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
  - $x \cdot e = x$
  - $e \cdot x = x$
  - $x \cdot x' = e$
- Groups can be defined in different ways, including by **presentations** $\langle x_1, \ldots, x_n; r_1, \ldots, r_m \rangle$, where $x_1, \ldots, x_n$ are *generators* and $r_1, \ldots r_m$ are *relators*

Groups and their presentations

- Groups are algebraic structures which satisfy the following axioms
  - $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
  - $x \cdot e = x$
  - $e \cdot x = x$
  - $x \cdot x' = e$
- Groups can be defined in different ways, including by **presentations** $\langle x_1, \ldots, x_n; r_1, \ldots, r_m \rangle$, where $x_1, \ldots, x_n$ are *generators* and $r_1, \ldots r_m$ are *relators*
- Intuitively, the presentation above defines a group the elements of which are words in the alphabet $x_1, \ldots, x_n, x'_1, \ldots, x'_n$ taken up to the equivalence defined by $r_1 = e, \ldots, r_m = e$

Trivial group presentations

- $\langle a, b \mid ab, b \rangle$ (trivial example of the trivial group presentation)

Trivial group presentations

- $\langle a, b \mid ab, b\rangle$ (trivial example of the trivial group presentation)
- $\langle a, b \mid abab'a'b', aaab'b'b'b'$ (not so trivial example of the trivial groups presentation)

For a group presentation $\langle x_1, \ldots, x_n; r_1, \ldots r_m \rangle$ with generators x_i, and relators r_j, consider the following transformations.

- AC1 Replace some r_i by r_i^{-1}.
- AC2 Replace some r_i by $r_i \cdot r_j$, $j \neq i$.
- AC3 Replace some r_i by $w \cdot r_i \cdot w^{-1}$ where w is any word in the generators.

Andrews-Curtis Conjecture

- Two presentations g and g' are called *Andrews-Curtis equivalent (AC-equivalent)* if one of them can be obtained from the other by applying a finite sequence of transformations of the types (AC1) - (AC3).
- A group presentation $g = \langle x_1, \ldots, x_n; r_1, \ldots r_m \rangle$ is called *balanced* if $n = m$, that is a number of generators is the same as a number of relators. Such n we call a *dimension* of g and denote by $Dim(g)$.

Conjecture (1965)

if $\langle x_1, \ldots, x_n; r_1, \ldots r_n \rangle$ is a balanced presentation of the trivial group it is AC-equivalent to the trivial presentation $\langle x_1, \ldots, x_n; x_1, \ldots x_n \rangle$.

Trivial Example

- $\langle a, b \mid ab, b\rangle \to \langle a, b \mid ab, b^{-1}\rangle \to \langle a, b \mid a, b^{-1}\rangle \to \langle a, b \mid a, b\rangle$

AC-conjecture: short profile

- AC-conjecture is open

AC-conjecture: short profile

- AC-conjecture is open
- AC-conjecture may well be false (prevalent opinion of experts?)

AC-conjecture: short profile

- AC-conjecture is open
- AC-conjecture may well be false (prevalent opinion of experts?)
- Series of potential counterexamples; smallest for which simplification is unknown is AK-3: $\langle x, y | xyxy^{-1}x^{-1}y^{-1}, x^3y^{-4} \rangle$

AC-conjecture: short profile

- AC-conjecture is open
- AC-conjecture may well be false (prevalent opinion of experts?)
- Series of potential counterexamples; smallest for which simplification is unknown is AK-3: $\langle x, y | xyxy^{-1}x^{-1}y^{-1}, x^3y^{-4} \rangle$
- How to find simplifications, algorithmically?

AC-conjecture: short profile

- AC-conjecture is open
- AC-conjecture may well be false (prevalent opinion of experts?)
- Series of potential counterexamples; smallest for which simplification is unknown is AK-3: $\langle x, y | xyxy^{-1}x^{-1}y^{-1}, x^3y^{-4} \rangle$
- How to find simplifications, algorithmically?
- If a simplification exists, it could be found by the exhaustive search/total enumeration (iterative deepening)
- The issue: simplifications could be very long (Bridson 2015; Lishak 2015)

Search of trivializations and elimination of counterexamples

- Genetic search algorithms (Miasnikov 1999; Swan et al. 2012)
- Breadth-First search (Havas-Ramsay, 2003; McCaul-Bowman, 2006)
- Todd-Coxeter coset enumeration algorithm (Havas-Ramsay,2001)
- Generalized moves and strong equivalence relations (Panteleev-Ushakov, 2016)
- ...

Search of trivializations and elimination of counterexamples

- Genetic search algorithms (Miasnikov 1999; Swan et al. 2012)
- Breadth-First search (Havas-Ramsay, 2003; McCaul-Bowman, 2006)
- Todd-Coxeter coset enumeration algorithm (Havas-Ramsay,2001)
- Generalized moves and strong equivalence relations (Panteleev-Ushakov, 2016)
- ...

Our approach: apply generic automated reasoning instead of specialized algorithms
Our Claim: generic automated reasoning is (very) competitive

ACT rewriting system, dim =2

Equational theory of groups T_G:

- $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
- $x \cdot e = x$
- $e \cdot x = x$
- $x \cdot r(x) = e$

For each $n \geq 2$ we formulate a term rewriting system modulo T_G, which captures AC-transformations of presentations of dimension n. For an alphabet $A = \{a_1, a_2\}$ a term rewriting system ACT_2 consists the following rules:

$$
\begin{array}{rl}
\text{R1L} & f(x,y) \to f(r(x),y)) \\
\text{R1R} & f(x,y) \to f(x,r(y)) \\
\text{R2L} & f(x,y) \to f(x \cdot y, y) \\
\text{R2R} & f(x,y) \to f(x, y \cdot x) \\
\text{R3L}_i & f(x,y) \to f((a_i \cdot x) \cdot r(a_i), y) \text{ for } a_i \in A, i = 1,2 \\
\text{R3R}_i & f(x,y) \to f(x, (a_i \cdot y) \cdot r(a_i)) \text{ for } a_i \in A, i = 1,2
\end{array}
$$

AC-transformations as rewriting modulo group theory

The rewrite relation $\rightarrow_{ACT/G}$ for ACT modulo theory T_G:
$t \rightarrow_{ACT/G} s$ iff there exist $t' \in [t]_G$ and $s' \in [s]_G$ such that $t' \rightarrow_{ACT} s'$.

Reduced ACT_2

Reduced term rewriting system $rACT_2$ consists of the following rules:

$$\begin{array}{rl} \text{R1L} & f(x,y) \to f(r(x),y)) \\ \text{R2L} & f(x,y) \to f(x \cdot y, y) \\ \text{R2R} & f(x,y) \to f(x, y \cdot x) \\ \text{R3L}_i & f(x,y) \to f((a_i \cdot x) \cdot r(a_i), y) \text{ for } a_i \in A, i = 1,2 \end{array}$$

Proposition

Term rewriting systems ACT_2 and $rACT_2$ considered modulo T_G are equivalent, that is $\to^*_{ACT_2/G}$ and $\to^*_{rACT_2/G}$ coincide.

Proposition

For ground t_1 and t_2 we have $t_1 \to^*_{ACT_2/G} t_2 \Leftrightarrow t_2 \to^*_{ACT_2/G} t_1$, that is $\to^*_{ACT_2/G}$ is symmetric.

Denote by E_{ACT_2} an equational theory $T_G \cup rACT^=$ where $rACT^=$ includes the following axioms (equality variants of the above rewriting rules):

$$
\begin{array}{rl}
\text{E-R1L} & f(x,y) = f(r(x),y)) \\
\text{E-R2L} & f(x,y) = f(x \cdot y, y) \\
\text{E-R2R} & f(x,y) = f(x, y \cdot x) \\
\text{E-R3L}_i & f(x,y) = f((a_i \cdot x) \cdot r(a_i), y) \text{ for } a_i \in A, i = 1,2
\end{array}
$$

Proposition

For ground terms t_1 and t_2 $t_1 \rightarrow^*_{ACT_2/G} t_2$ iff $E_{ACT_2} \vdash t_1 = t_2$

A variant of the equational translation: replace the axioms $\mathbf{E-R3L_i}$ by “non-ground" axiom $\mathbf{E-RLZ}$: $f(x,y) = f((z \cdot x) \cdot r(z), y)$

Implicational Translation

Denote by I_{ACT_2} the first-order theory $T_G \cup rACT_2^{\rightarrow}$ where $rACT_2^{\rightarrow}$ includes the following axioms:

I-R1L $R(f(x,y)) \rightarrow R(f(r(x),y)))$

I-R2L $R(f(x,y)) \rightarrow R(f(x \cdot y, y))$

I-R2R $R(f(x,y)) \rightarrow R(f(x, y \cdot x))$

I-R3L$_i$ $R(f(x,y)) \rightarrow R(f((a_i \cdot x) \cdot r(a_i), y))$ for $a_i \in A, i = 1, 2$

Proposition

For ground terms t_1 and t_2 $t_1 \rightarrow^*_{ACT_2/G} t_2$ iff $I_{ACT_2} \vdash R(t_1) \rightarrow R(t_2)$

- An equational translation for $n = 3$ (“non-ground” variant):
  $f(x, y, z) = f(r(x), y, z)$ $f(x, y, z) = f(x, r(y), z)$
  $f(x, y, z) = f(x, y, r(z))$ $f(x, y, z) = f(x \cdot y, y, z)$
  $f(x, y, z) = f(x \cdot z, y, z)$ $f(x, y, z) = f(x, y \cdot x, z)$
  $f(x, y, z) = f(x, y \cdot z, z)$ $f(x, y, z) = f(x, y, z \cdot x)$
  $f(x, y, z) = f(x, y, z \cdot y)$
  $f(x, y, z) = f((v \cdot x) \cdot r(v), y, z)$
  $f(x, y, z) = f(x, (v \cdot y) \cdot r(v), z)$ $f(x, y, z) = f(x, y, (v \cdot z) \cdot r(v))$.

For any pair of presentations p_1 and p_2,
to establish whether they are AC-equivalent one can formulate and try to solve first-order theorem proving problems

- $E_{ACT_n} \vdash t_{p_1} = t_{p_2}$, or
- $I_{ACT_n} \vdash R(t_{p_1}) \to R(t_{p_2})$

OR, theorem disproving problems

- $E_{ACT_n} \nvdash t_{p_1} = t_{p_2}$, or
- $I_{ACT_n} \nvdash R(t_{p_1}) \to R(t_{p_2})$

Automated Reasoning for AC conjecture exploration

For any pair of presentations p_1 and p_2,
to establish whether they are AC-equivalent one can formulate and try to solve first-order theorem proving problems

- $E_{ACT_n} \vdash t_{p_1} = t_{p_2}$, or
- $I_{ACT_n} \vdash R(t_{p_1}) \to R(t_{p_2})$

OR, theorem disproving problems

- $E_{ACT_n} \not\vdash t_{p_1} = t_{p_2}$, or
- $I_{ACT_n} \not\vdash R(t_{p_1}) \to R(t_{p_2})$

Our proposal: apply automated reasoning: ATP and finite model building.

Elimination of potential counterexamples

- **Known cases:** We have applied automated theorem proving using Prover9 prover to confirm that all cases eliminated as potential counterexamples in all known literature can be eliminated by our method too.

Theorem Proving for AC-Simplifications (cont.)

New cases (from Edjvet-Swan, 2005-2010):

T14 $\langle a, b \mid ababABB, babaBAA \rangle$
T28 $\langle a, b \mid aabbbbABBBB, bbaaaaBAAAA \rangle$
T36 $\langle a, b \mid aababAABB, bbabaBBAA \rangle$
T62 $\langle a, b \mid aaabbAbABBB, bbbaaBaBAAA \rangle$
T74 $\langle a, b \mid aabaabAAABB, bbabbaBBBAA \rangle$

T16 $\langle a, b, c \mid ABCacbb, BCAbacc, CABcbaa \rangle$
T21 $\langle a, b, c \mid ABCabac, BCAbcba, CABcacb \rangle$
T48 $\langle a, b, c \mid aacbcABCC, bbacaBCAA, ccbabCABB \rangle$
T88 $\langle a, b, c \mid aacbAbCAB, bbacBcABC, ccbaCaBCA \rangle$
T89 $\langle a, b, c \mid aacbcACAB, bbacBABC, ccbaCBCA \rangle$

T96 $\langle a, b, c, d \mid adCADbc, baDBAcd, cbACBda, dcBDCab \rangle$
T97 $\langle a, b, c, d \mid adCAbDc, baDBcAd, cbACdBa, dcBDaCb \rangle$ [ICMS 2018]

AC-trivialization for **T16**

$$\langle ABCacbb, BCAbacc, CABcbaa\rangle$$

$$\xrightarrow{x,y,z\to x,y,azA} \langle ABCacbb, BCAbacc, aCABcba\rangle$$

$$\xrightarrow{x,y,z\to x,y,zx} \langle ABCacbb, BCAbacc, aCABacbb\rangle$$

$$\xrightarrow{x,y,z\to x,y,bzB} \langle ABCacbb, BCAbacc, baCABacb\rangle$$

$$\xrightarrow{x,y,z\to x,y,zy} \langle ABCacbb, BCAbacc, bac\rangle$$

$$\xrightarrow{x,y,z\to x,y,czC} \langle ABCacbb, BCAbacc, cba\rangle$$

$$\xrightarrow{x,y,z\to x',y,z} \langle BBCAcba, BCAbacc, cba\rangle$$

$$\xrightarrow{x,y,z\to x,y,z'} \langle BBCAcba, BCAbacc, ABC\rangle$$

$$\xrightarrow{x,y,z\to xz,y,z} \langle BBCA, BCAbacc, ABC\rangle$$

$$\xrightarrow{x,y,z\to x',y,z} \langle acbb, BCAbacc, ABC\rangle \xrightarrow{x,y,z\to x,y,z'}$$

$$\langle acbb, BCAbacc, cba\rangle$$

$$\xrightarrow{x,y,z\to x,y,azA} \langle acbb, BCAbacc, acb\rangle \xrightarrow{x,y,z\to x,y,z'}$$

$$\langle acbb, BCAbacc, BCA\rangle$$

$$\xrightarrow{x,y,z\to x,y,zx} \langle acbb, BCAbacc, b\rangle \xrightarrow{x,y,z\to x,y,z'} \langle acbb, BCAbacc, B\rangle$$

$$\xrightarrow{x,y,z\to xz,y,z} \langle acb, BCAbacc, B\rangle \xrightarrow{x,y,z\to xz,y,z} \langle ac, BCAbacc, B\rangle$$

AC-trivialization for **T16** (cont.)

$$\xrightarrow{x,y,z\to x,y',z} \langle ac, CCABacb, B\rangle \xrightarrow{x,y,z\to x,yz,z} \langle ac, CCABac, B\rangle$$

$$\xrightarrow{x,y,z\to x,y',z} \langle ac, CAbacc, B\rangle \xrightarrow{x,y,z\to x,y,z'} \langle ac, CAbacc, b\rangle$$

$$\xrightarrow{x,y,z\to x',y,z} \langle CA, CAbacc, b\rangle$$

$$\xrightarrow{x,y,z\to x,yx,z} \langle CA, CAbacA, b\rangle \xrightarrow{x,y,z\to x,y',z} \langle CA, aCABac, b\rangle$$

$$\xrightarrow{x,y,z\to x,yx,z} \langle CA, aCAB, b\rangle \xrightarrow{x,y,z\to x,yz,z} \langle CA, aCA, b\rangle$$

$$\xrightarrow{x,y,z\to x',y,z} \langle ac, aCA, b\rangle \xrightarrow{x,y,z\to x,yx,z} \langle ac, a, b\rangle$$

$$\xrightarrow{x,y,z\to x,y',z} \langle ac, A, b\rangle \xrightarrow{x,y,z\to x,yx,z} \langle ac, c, b\rangle$$

$$\xrightarrow{x,y,z\to x,y',z} \langle ac, C, b\rangle \xrightarrow{x,y,z\to xy,y,z} \langle a, C, b\rangle$$

$$\xrightarrow{x,y,z\to x,yz,z} \langle a, Cb, b\rangle \xrightarrow{x,y,z\to x,y',z} \langle a, Bc, b\rangle$$

$$\xrightarrow{x,y,z\to x,y,zy} \langle a, Bc, c\rangle \xrightarrow{x,y,z\to x,y,z'} \langle a, Bc, C\rangle$$

$$\xrightarrow{x,y,z\to x,yz,z} \langle a, B, C\rangle \xrightarrow{x,y,z\to x,y,z'} \langle a, B, c\rangle$$

$$\xrightarrow{x,y,z\to x,y',z} \langle a, b, c\rangle$$

What about automated disproving?

Proposition

To simplify AK-3 (if at all it is possible) one really needs conjugation with both generators a and b.

Mace4 finite model builder finds countermodels of sizes 12 and 6 for the cases where either of the conjugation rules is omitted.

What about automated disproving?

Proposition

To simplify AK-3 (if at all it is possible) one really needs conjugation with both generators a and b.

Mace4 finite model builder finds countermodels of sizes 12 and 6 for the cases where either of the conjugation rules is omitted.

Can we disprove AC-conjecture by building a finite countermodel witnessing non-trivialization for one of the open cases (e.g. AK-3)?

What about automated disproving?

Proposition

To simplify AK-3 (if at all it is possible) one really needs conjugation with both generators a and b.

Mace4 finite model builder finds countermodels of sizes 12 and 6 for the cases where either of the conjugation rules is omitted.

Can we disprove AC-conjecture by building a finite countermodel witnessing non-trivialization for one of the open cases (e.g. AK-3)?

No, unfortunately (Borovik et al, *The Finitary Andrews-Curtis Conjecture*, 2005)

Automorphic Moves

(Panteleev-Ushakov, 2016): add automorphisms of F_2 to the set of AC-moves

AT1 Replace $\bar{r}$ by $\phi_1(\bar{r})$, where $\phi_1(\ldots)$ is an automorphism defined by $a \mapsto a$ and $b \mapsto b^{-1}$.

AT2 Replace $\bar{r}$ by $\phi_2(\bar{r})$, where $\phi_2(\ldots)$ is an automorphism defined by $a \mapsto a$ and $b \mapsto b * a$.

AT3 Replace $\bar{r}$ by $\phi_3(\bar{r})$, where $\phi_3(\ldots)$ is an automorphism defined by $a \mapsto b$ and $b \mapsto a$.

Automorphic Moves: known properties

Adding Automorphic moves to AC does not increase the sets of reachable presentations when:

- applied to AC-trivializable presentations (easy to see);
- applied to Akbulut-Kirby presentations $AK(n), n \geq 3$ (not known to be trivializable) (Panteleev-Ushakov, 2016)

Adding Automorprhic moves to AC does not increase the sets of reachable presentations when:

- applied to AC-trivializable presentations (easy to see);
- applied to Akbulut-Kirby presentations $AK(n), n \geq 3$ (not known to be trivializable) (Panteleev-Ushakov, 2016)

The general case was left open in Op.cit.:

> *It is not known if adding these transformations to AC-moves results in an equivalent system of transformations or not ...*

AR for automorphic moves

We answer the question negatively and show that adding *any* AT move to AC transformations does indeed lead to a non-equivalent system of transformations:

Theorem

A group presentation $g = \langle a, b \mid aba, bba \rangle$ *is not AC -equivalent to either of*

- $g_1 = \langle a, b \mid \phi_1(aba), \phi_1(bba) \rangle \equiv \langle a, b \mid ab^{-1}a, b^{-1}b^{-1}a \rangle$
- $g_2 = \langle a, b \mid \phi_2(aba), \phi_2(bba) \rangle \equiv \langle a, b \mid abaa, babaa \rangle$

A group presentation $g' = \langle a, b \mid aaba, bba \rangle$ *is not AC -equivalent to*

- $g_3 = \langle a, b \mid \phi_3(aaba), \phi_3(bba) \rangle \equiv \langle a, b \mid bbab, aab \rangle$

Proof using AR

Apply equational translation and show that $\tilde{E}_{ACT_2} \not\vdash t_g = t_{g_i}$ $i = 1, 2$ and $\tilde{E}_{ACT_2} \not\vdash t_{g'} = t_{g_3}$.

Mace4 has found the following countermodels

1) For
$\tilde{E}_{ACT_2} \not\vdash f((a * b) * a, (b * b) * a) = f((a * r(b)) * a, (r(b) * r(b)) * a)$:

```
interpretation( 3, [number = 1,seconds = 0], [
    function(*(_,_), [
        2,0,1,
        0,1,2,
        1,2,0]),
    function(a, [0]),
    function(b, [0]),
    function(e, [1]),
    function(r(_), [2,1,0]),
    function(f(_,_), [
        0,0,0,
        0,1,0,
        0,0,0])]).
```

Proof using AR (cont.)

2) For $\tilde{E}_{ACT_2} \not\vdash f((a*(b*a))*a, ((b*a)*(b*a))*a)$: the same as above.

3) For $\tilde{E}_{ACT_2} \not\vdash f((a*(a*b))*a, (b*b)*a) =$
$f((a*(a*b))*a, (b*b)*a) = f((b*(b*a))*b, (a*a)*b)$:

```
interpretation( 5, [number = 1,seconds = 0], [
    function(*(_,_), [
        4,3,0,2,1,
        3,0,1,4,2,
        0,1,2,3,4,
        2,4,3,1,0,
        1,2,4,0,3]),
    function(a, [0]),
    function(b, [1]),
    function(e, [2]),
    function(r(_), [3,4,2,0,1]),
  ....
```

PU algorithmic approach vs AR

(Panteleev-Ushakov, 2016):

- Powerful algorithmic approach to AC-transformations based on generalized moves and strong equivalence relations;
- 12 novel AC-trivializations for presentations:

```
(XyyxYYY,xxYYYXYxYXYY) (XyyxYYY,xxyyyXYYXyxY)
(XyyxYYY,xxYXyxyyyXY) (XyyxYYY,xxYXyXyyxyy)
...
```

PU algorithmic approach vs AR

(Panteleev-Ushakov, 2016):

- Powerful algorithmic approach to AC-transformations based on generalized moves and strong equivalence relations;
- 12 novel AC-trivializations for presentations:

  `(XyyxYYY,xxYYYXYxYXYY)` `(XyyxYYY,xxyyyXYYXyxY)`
  `(XyyxYYY,xxYXyxyyyXY)` `(XyyxYYY,xxYXyXyyxyy)`
  ...

  All confirmed by our AR method!
- 16 presentations are shown to be AC-equivalent to F_2 automorphic images:

  `(xxxyXXY,xyyyyXYYY)` `(xxxyXXY,xyyyXYYYY)`
  `(xyyyXYY,xxxyyXXY)` `(xxxyXXY,xxyyyXYY)`
  ...

PU algorithmic approach vs AR

(Panteleev-Ushakov, 2016):

- Powerful algorithmic approach to AC-transformations based on generalized moves and strong equivalence relations;
- 12 novel AC-trivializations for presentations:

  `(XyyxYYY,xxYYYXYxYXYY) (XyyxYYY,xxyyyXYYXyxY)`
  `(XyyxYYY,xxYXyxyyyXY) (XyyxYYY,xxYXyXyyxyy)`
  ...

  All confirmed by our AR method!
- 16 presentations are shown to be AC-equivalent to F_2 automorphic images:

  `(xxxyXXY,xyyyyXYYY) (xxxyXXY,xyyyXYYYY)`
  `(xyyyXYY,xxxyyXXY) (xxxyXXY,xxyyyXYY)`
  ...

  Our AR method failed for all cases!

Conclusion

- Automated Proving and Disproving is an interesting and powerful approach to AC-conjecture exploration;
- Source of interesting challengeing problems for ATP/ATD;
- Can ML help to guide the proofs?

Conclusion

- Automated Proving and Disproving is an interesting and powerful approach to AC-conjecture exploration;
- Source of interesting challengeing problems for ATP/ATD;
- Can ML help to guide the proofs?

Thank you!

Time to prove simplifications

	T14	T28	T36	T62	T74	T16	T21	T48	T88	T89	T96	97
Dim	2	2	2	2	2	3	3	3	3	3	4	4
Equational	6.02s	6.50s	7.18s	24.34s	57.17s	12.87s	11.98s	34.63s	57.69s	17.50s	114.05s	115.10s
Implicational	1.57s	2.46s	1.34s	22.50s	6.29s	1.61s	1.45s	2.17s	1.97s	2.14s	102.34s	89.65s
Implicational GC	t/o	t/o	t/o	t/o	t/o	3.76s	1.61s	t/o	0.86s	0.75s	t/o	t/o

“t/o” stands for timeout in 200s; “GC” means encoding with ground conjugation rules; all other encodings are with non-ground conjugation rules.