

Automated reasoning in algebra

David Stanovský

Charles University in Prague
Czech Republic

stanovsk@karlin.mff.cuni.cz
<http://www.karlin.mff.cuni.cz/~stanovsk>

Linz, November 2007

- Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

- Finding a proof: Prover9, Waldmeister, E, Vampire, SPASS, ...
- Finding a finite model: Paradox, Mace4, Darwin, ...

- Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

- Finding a proof: Prover9, Waldmeister, E, Vampire, SPASS, ...
- Finding a finite model: Paradox, Mace4, Darwin, ...

- Automated theory building

- Proof verification

- Isabelle, HOL, Coq
- Mizar

- etc.

Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

Main troubles:

- undecidable, exponential search space
- first order within a theory / in ZFC

Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

Main troubles:

- undecidable, exponential search space
- first order within a theory / in ZFC

How could it possibly help *YOU*:

- Checking conjectures on small models.
- Find a proof, look at hints, find a better proof by hand.
- Find a proof, translate it to human language.
- Exhaustive search.

Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

Main troubles:

- undecidable, exponential search space
- first order within a theory / in ZFC

How could it possibly help *YOU*:

- Checking conjectures on small models.
- Find a proof, look at hints, find a better proof by hand.
- Find a proof, translate it to human language.
- Exhaustive search.

When AR tools can overtake mathematician's brain:

- not-really-well-understood equations
- find complicated syntactic proofs
- quickly find small models, without their real understanding

Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

Algorithms and implementations:

- Resolution with paramodulation: Vampire, E, Otter/Prover9, SPASS
- Term rewriting: Waldmeister
- Tableaux: few experimental ones
- few other experimental techniques

Automated theorem proving

INPUT: finite set of first order formulas

OUTPUT: Satisfiable / Unsatisfiable / I don't know (Timeout)

Algorithms and implementations:

- Resolution with paramodulation: Vampire, E, Otter/Prover9, SPASS
- Term rewriting: Waldmeister
- Tableaux: few experimental ones
- few other experimental techniques

Benchmarks: <http://www.tptp.org>

- TPTP library
- CASC competition
- TPTP \neq algebraic problems — algebraists use mostly Otter/Prover9

Milestones:

- 1996, W. McCune: Robbins algebras are Boolean algebras
- 1996, K. Kunen: Moufang quasigroups are loops
- since early 90's: short axioms for various theories
- since early 00's, M. Kinyon, JD Phillips, P. Vojtěchovský: everyday use in loop theory
- recent years, B. Fitelson, M. Spinks, etc.: algebraic logic

Robbins' problem

(Huntington, 1933) Short axioms for Boolean algebras:

$$x + y = y + x, \quad (x + y) + z = x + (y + z), \\ (x' + y)' + (x' + y')' = x.$$

(Robbins, 1934) Shorter axioms, conjectured to axiomatize BA's:

$$x + y = y + x, \quad (x + y) + z = x + (y + z), \\ ((x + y)' + (x + y')')' = x.$$

(Winker, 1979) Sufficient to prove that

$$\text{Robbins} \vdash (\exists A)(\exists B) (A + B)' = A'$$

Confirmed by EQP prover by McCune in 1996, reported in NY Times (!)

Single axioms

(McCune, 1993) The *shortest* axiom for *abelian groups*:

$$((x * y) * z) * (x * z)' = y$$

(Kunen, 1992; McCune, 1993) Short single axioms for *groups*:

3 variables: $((z * (x * y))' * (z * y')) * (y' * y)' = x$

4 variables: $y * (z * (((w * w') * (x * z))' * y))' = x$

(McCune, Padmanabhan, Veroff, 2002) A short axiom for *lattices*:

$$(((y \vee x) \wedge x) \vee (((z \wedge (x \vee x)) \vee (u \wedge x)) \wedge v)) \wedge (((w \vee x) \wedge (r \vee x)) \vee s) = x$$

(McCune, Veroff, Fitelson, Harris, Feist, Wos, 2002)

A *shortest* axiom for *Boolean algebras* in terms of Sheffer stroke:

$$((x|y)|z)|(x|((x|z)|x)) = z$$

Moufang quasigroups

Quasigroup = latin square = (G, \cdot) , all translations are permutations

Loop = quasigroup with a unit = non-associative group

$$x \setminus (x \cdot y) = y, \quad x \cdot (x \setminus y) = y, \quad (y/x) \cdot x = y, \quad (y \cdot x)/x = y$$
$$x \cdot 1 = 1 \cdot x = x$$

Moufang identity (weak associativity):

$$((x \cdot y) \cdot x) \cdot z = x \cdot (y \cdot (x \cdot z))$$

Is every Moufang quasigroup a loop?

Proved with McCune's Otter by Kenneth Kunen in 1996.

(More results of this type later.)

Other results on quasigroups and loops

(Kinyon, Kunen, Phillips) Diassociative A-loops are Moufang

- diassociative = 2-generated subloops are groups
- A-loop = inner mappings are automorphisms

(Kinyon, Phillips) Trimedial quasigroups

- quasigroups where 3-generated subquasigroups satisfy $xy \cdot uv = xu \cdot yv$
- equational base: $x \cdot yz = (x/x)y \cdot xz$, $zy \cdot x = zx \cdot y(x \setminus x)$

(Phillips, Vojtěchovský) Varieties of qgrps/loops of Bol-Moufang type

- equations of type $abcd = abcd$: determine the \vdash relation

(Nagy) Finite simple Bol loops

- For many years, it used to be Open Problem #1 in loop theory, whether there is a non-Moufang one
- found by GAP, size 24

Distributive groupoids are symmetric-by-medial

- In groupoids,

$$x * yz = xy * xz, \quad xy * z = xz * yz$$

implies

$$(xy * zu) * ((xy * zu) * (xz * yu)) = xz * yu$$

$$(xy * zu) * (xz * yu) = (xz * yu) * (xy * zu)$$

- Semantical meaning for idempotent groupoids:
there is a congruence α such that \mathbf{G}/α is medial and all blocks are symmetric subgroupoids.
- A higher order proof was known, but difficult.
- Prover9 found a relatively short equational proof.

Axioms of biquandles

- A *birack* is an algebra $(A, \circ, *, \backslash_{\circ}, \backslash_*)$ satisfying

$$x \circ (y \circ z) = (x \circ y) \circ ((y * x) \circ z)$$

$$x * (y * z) = (x * y) * ((y \circ x) * z)$$

$$((x * y) \circ z) * (y \circ x) = ((x \circ z) * y) \circ (z * x)$$

$$x \circ (x \backslash_{\circ} y) = y, \quad x \backslash_{\circ} (x \circ y) = y$$

$$x * (x \backslash_* y) = y, \quad x \backslash_* (x * y) = y$$

- In every birack,

$$(x \backslash_{\circ} x) \backslash_* (x \backslash_{\circ} x) = x \quad \Leftrightarrow \quad (x \backslash_* x) \backslash_{\circ} (x \backslash_* x) = x.$$

(Such algebras are called *biquandles*, are used in knot theory.)

My results III. – Linear theories of groupoids

Problem:

- Describe **-linear theories* = equational theories where every term is equivalent to a unique *linear* term.
- *Linear term* = each variable at most once
 - 1 variable: x
 - 2 variables: x, y, xy, yx
 - 3 variables: $x, y, z, xy, yx, xz, zx, yz, zy, x(yz), (xy)z, \dots$

Auxilliary problem:

- Describe *n-linear theories* = every term *in $\leq n$ variables* is equivalent to a unique linear term.
 - Determined by its n -generated free groupoid.

My results III. – Linear theories of groupoids

Problem:

- Describe **-linear theories* = equational theories where every term is equivalent to a unique *linear* term.
- *Linear term* = each variable at most once
 - 1 variable: x
 - 2 variables: x, y, xy, yx
 - 3 variables: $x, y, z, xy, yx, xz, zx, yz, zy, x(yz), (xy)z, \dots$

Auxilliary problem:

- Describe *n-linear theories* = every term *in $\leq n$ variables* is equivalent to a unique linear term.
 - Determined by its n -generated free groupoid.

Procedure:

- 1 Find all *2-linear* and *3-linear* theories.
- 2 Prove that only 3 of them can be extended to a *4-linear* one.
- 3 **-linear* theory is generated by its 4-generated free groupoid.
- 4 Each of the three 3-linear theories extends to at most one **-linear*.
- 5 Explicit construction of three **-linear* theories.

My results III. – Search for 1,2,3,4-linear theories

Facts:

- Sizes of free groupoids: 1, 4, 21, 184.
- About $1/(\# \text{ of vars.})$ of the mlt table determines the free groupoid.
- n -generated = extension of $(n - 1)$ -generated.

Solution:

- A Perl script prepares all possible completions of the mlt table.
- For each of them, Otter checks whether the corresponding theory collapses different linear terms.

Results:

- 2-generated free groupoids:
Appeared earlier in literature, about *one minute* to compute them.
- 2-generated free groupoids:
Several days by hand, about *two hours* to compute them.
- 4 of 7 3-generated free groupoids don't have 4-linear extension:
Quite easy by hand, *several weeks* to compute this fact.

More advanced projects = future of ATP?

(A random choice of recent projects I found interesting.)

- Search for isomorphism/isotopy *invariants* for loops
 - Paradox: generates models
 - HR: searches for interesting formulas valid in a given model
 - ATP's: prove that invariants cover all models of given size
- *MPTP*: automated reasoning in ZFC
 - Problems for ATP's based on the Mizar library of formalized mathematics
 - MPTP \$100 challenge: automated proof of Bolzano-Weierstraß theorem
- *Malarea*: machine learning in service of automated reasoning
 - Reasoning in large theories (like ZFC with some math background)
 - Problem: Which axioms are useful for given problem?
Machine learning based on syntactical analysis of given conjectures.
 - Relatively succesful on the MPTP challenge

<http://www.karlin.mff.cuni.cz/~stanovsk>

<http://www.tptp.org>

<http://www.prover9.org>

<http://www.cs.unm.edu/~veroff/ADAM/>