Automated Theorem Proving in Loop Theory

JD Phillips and David Stanovský

Wabash College, Crawfordsville, IN
Charles University in Prague, Czech Republic

phillipj@wabash.edu
stanovsk@karlin.mff.cuni.cz
http://www.karlin.mff.cuni.cz/˜stanovsk

ESARM, Birmingham 2008

[The authors] demonstrate that (contrary to the view amongst some in AR), provided a sufficiently effective AR tool is available, there are some mathematicians who will indeed use such a tool.

— anonymous referee

[The authors] demonstrate that (contrary to the view amongst some in AR), provided a sufficiently effective AR tool is available, there are some mathematicians who will indeed use such a tool.

— anonymous referee

This talk

- is about solving open problems by first order automated theorem provers
- is *not* about formal verification or theory formation

Automated theorem proving in mathematics

(Almost) useless!

Automated theorem proving in mathematics

(Almost) useless!

- undecidable, slow
- first order problems within a given theory

Sometimes useful...

- quickly checking easy conjectures
  (typically, find a small counterexample, without its real understanding)
- not really well understood equations
- find complicated syntactic proofs
- exhaustive search

Automated theorem proving in algebra

Some examples:

- short axioms for various theories (since early 90's)
- Robbins problem (1996)
- loop theory (since 1996)
- algebraic logic (last couple years)

My older results:

- some properties of selfdistributive algebras
- classification of free algebras in 4-linear theories

Automated theorem proving in loop theory

Milestones:

- 1996, K. Kunen: first use (Moufang quasigroups are loops)
- 2001, Kinyon and Phillips learned to use Otter
- tutorial at Loops'04, ATP becomes a standard tool
- since 2008, more provers in use

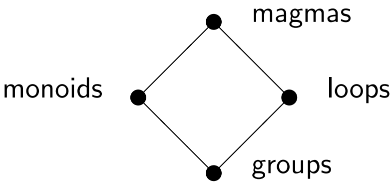
Achievments:

- several longstanding open problems
- significant new results in various projects
- 21 papers, where results were obtained with assistance of ATP

Techniques:

- Otter, Prover9 (until 2007), Waldmeister
- parameter setting, *hints strategy*
- proofs always translated

Two paths from magmas to groups



Magma $= (A, *, 1)$, where $x * 1 = 1 * x = x$
Monoid $=$ magma & associative
Loop $=$ magma & for every a, b there are unique solutions of

$$a * x = b, \quad y * a = b$$

Group $=$ magma with both properties

Loops

Equational definition:

- language: $\cdot, /, \backslash, 1$
- axioms:

$$x1 = 1x = x$$

$$x\backslash(xy) = y, \quad x(x\backslash y) = y, \quad (yx)/x = y, \quad (y/x)x = y$$

Loops

Equational definition:

- language: $\cdot, /, \backslash, 1$
- axioms:

$$x1 = 1x = x$$

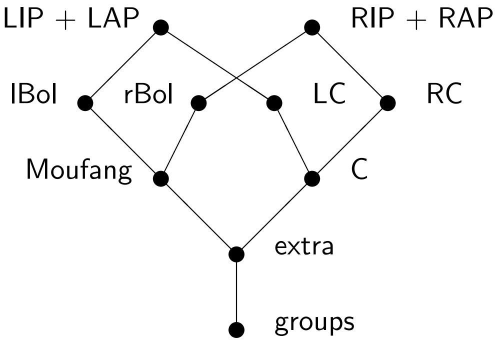
$$x\backslash(xy) = y, \quad x(x\backslash y) = y, \quad (yx)/x = y, \quad (y/x)x = y$$

Look at loop theory as generalization of group theory!

Selected topics:

- weak associativity
- inverses
- structural concepts
- tools (translations, subloops)

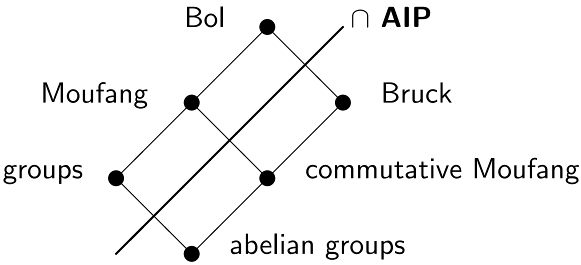
Weak associativity



$$x(y \cdot xz) = (x \cdot yx)z \qquad (\textit{left Bol})$$
$$x(y \cdot xz) = (xy \cdot x)z \qquad (\textit{Moufang})$$
$$x(y \cdot yz) = (x \cdot yy)z \qquad (\textit{LC})$$
$$x(y \cdot zx) = (xy \cdot z)x \qquad (\textit{extra})$$

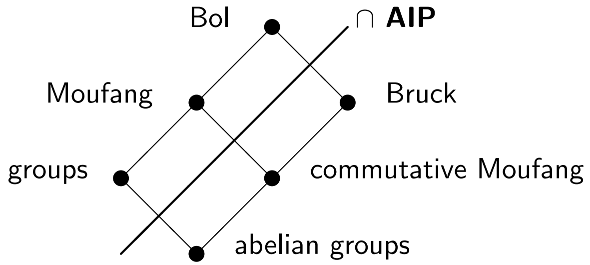
Inverse: x^{-1} such that $x^{-1}x = xx^{-1} = 1$ — may not exist!

AAIP: $(xy)^{-1} = y^{-1}x^{-1}$ *AIP:* $(xy)^{-1} = x^{-1}y^{-1}$



Inverse: x^{-1} such that $x^{-1}x = xx^{-1} = 1$ — may not exist!

AAIP: $(xy)^{-1} = y^{-1}x^{-1}$ *AIP:* $(xy)^{-1} = x^{-1}y^{-1}$



$$x^{-1} \cdot xy = y \qquad (LIP)$$

$$x \cdot xy = xx \cdot y \qquad (LAP)$$

Important subsets, subloops, ...

Commutant: $C(Q) = \{a \in Q : ax = xa, \forall x \in Q\}$

Nucleus: $N(Q) = N_\lambda(Q) \cap N_\mu(Q) \cap N_\rho(Q)$

$$N_\lambda(Q) = \{a \in Q : a \cdot xy = ax \cdot y, \forall x, y \in Q\}$$

$$N_\lambda(Q) = \{a \in Q : x \cdot ay = xa \cdot y, \forall x, y \in Q\}$$

$$N_\lambda(Q) = \{a \in Q : x \cdot ya = xy \cdot a, \forall x, y \in Q\}$$

Center: $Z(Q) = N(Q) \cap C(Q)$

The bigger these subsets are, the closer the loop is to (abelian) group.

Translations: $L(a) : a \mapsto ax$, $R(a) : a \mapsto xa$

Multiplication group: $Mlt(Q) = \langle L(a), R(a) : a \in Q \rangle$

Inner mapping group: $Inn(Q) = \{f \in Mlt(Q) : f(1) = 1\}$

Use:

- define concepts, e.g.
  - *normal subloop* = invariant under the action of $Inn(Q)$
- handle equational properties
- new problems, e.g.
  - to what extent $Mlt(Q)$ or $Inn(Q)$ determine properties of Q ?
  - *A-loop* = inner mappings are automorphisms

QPTP = Quasigroup problems for theorem provers

= a collection of results in loop theory obtained with assistance of ATP

- all 21 papers covered (1996–2008)
- selected 80 problems (68 equational)

Benchmarking (E, Prover9, Spass, Vampire, Waldmeister):

- 71 problems solved by at least one prover
- 38 problems solved by all provers

QPTP language

```
#assumptions:
<<loop
<<associative
x*x=1.
#goals:
<<commutative
```

QPTP language

```
#assumptions:
<<loop
<<associative
x*x=1.
#goals:
<<commutative
```

⟶ qptp2tptp ⟶

```
cnf(sos,axiom,mult(A,e) = A).
cnf(sos,axiom,mult(e,A) = A).
cnf(sos,axiom,mult(A,ld(A,B)) = B).
cnf(sos,axiom,ld(A,mult(A,B)) = B).
cnf(sos,axiom,mult(rd(A,B),B) = A).
cnf(sos,axiom,rd(mult(A,B),B) = A).
cnf(sos,axiom,mult(A,mult(B,C)) = mult(mult(A,B),C)).
cnf(sos,axiom,mult(A,A) = e).
cnf(goals,negated_conjecture,mult(op_a,op_b) != mult(op_b,op_a)).
```

(1996 K. Kunen) *Every Moufang quasigroup a loop.*

```
#assumptions:
<<quasigroup
<<Moufang1
#goals:
<<q_unit
```

(1996 K. Kunen) *Every Moufang quasigroup a loop.*

```
#assumptions:
<<quasigroup
<<Moufang1
#goals:
<<q_unit
```

What is existence of a unit?

- $\exists x \forall y \; xy = yx = y$
- $y(x/x) = y \;\&\; (x/x)y = y$
- $y(x\backslash x) = y \;\&\; (x\backslash x)y = y$

	E	Prover9	Spass	Vampire	Wm
Kun96a_1	56	75		258	x
Kun96a_1alt1	128	112		218	3
Kun96a_1alt2	9	68		238	3

(2001 Kinyon, Kunen, Phillips) *Diassociative A-loops are Moufang.*

Diassociative = satisfies all instances of associativity in 2 vars

- non-finitely based property
- in A-loops equivalent to IP property! *(manually)*

```
#assumptions:
<<loop
<<A
<<IP
<<Moufang234_imply_Moufang1
#goals:
<<Moufang1
```

(2001 Kinyon, Kunen, Phillips) *Diassociative A-loops are Moufang.*

Diassociative = satisfies all instances of associativity in 2 vars

- non-finitely based property
- in A-loops equivalent to IP property! *(manually)*

```
#assumptions:
<<loop
<<A
<<IP
<<Moufang234_imply_Moufang1
#goals:
<<Moufang1
```

	E	Prover9	Spass	Vampire	Wm
KKP02a_1	3023	26735			x
KKP02a_1alt1	848	36852		553	205
KKP02a_1alt2	848	35016		500	208
KKP02a_1alt3	1001	24832		550	213
KKP02a_1alt4	1018	24242		584	202

(2006 Aschbacher, Kinyon, Phillips)

In Bruck loops, elements of order 2^k commute with elements of odd order.

- can't prove for all integers
- can prove for some integers, then construct a general proof *(manually)*
- Application: a decomposition theorem for Bruck loops *(manually)*

```
#assumptions:
<<loop
<<Bruck
C*(C*(C*C))=1.
D*(D*D)=1.
#goals:
C*D=D*C.
```

(2006 Aschbacher, Kinyon, Phillips)

In Bruck loops, elements of order 2^k commute with elements of odd order.

- can't prove for all integers
- can prove for some integers, then construct a general proof *(manually)*
- Application: a decomposition theorem for Bruck loops *(manually)*

```
#assumptions:
<<loop
<<Bruck
C*(C*(C*C))=1.
D*(D*D)=1.
#goals:
C*D=D*C.
```

	E	Prover9	Spass	Vampire	Wm
2^2, 3	0	11	459	6	0
2^2, 3^2	16	1110			74
2^4, 3^2					

QPTP: overall performance

	E	Prover9	Spass	Vampire	Wm
proofs in 360s	53	46	31	44	46
proofs in 3600s	59	53	35	57	56
proofs in 86400s	62	61	39	60	59
timeouts	18	19	41	20	9

Main limitation of the benchmark: no *parameter setting*

- CASC strategy may not be the best for QPTP problems

QPTP: overall performance

	E	Prover9	Spass	Vampire	Wm
proofs in 360s	53	46	31	44	46
proofs in 3600s	59	53	35	57	56
proofs in 86400s	62	61	39	60	59
timeouts	18	19	41	20	9

Main limitation of the benchmark: no *parameter setting*

- CASC strategy may not be the best for QPTP problems

Future:

- play with settings
- merge with TPTP ($\rightarrow$ developers will do)
- more provers
- more domains

“Applications”

New theorems proved by Waldmeister!

- *Bruck loops with abelian* $Inn(L)$ *are nilpotent of class 2.*
- *Loops with abelian* $Inn(L)$ *of exponent 2 are abelian groups.*

Conclusions

- yes, we, mathematicians, want to use ATP
- ATPs can prove difficult theorems, just give them enough time
- a bit surprizingly, performance of ATPs on QPTP and UEQ TPTP is similar

Conclusions

- yes, we, mathematicians, want to use ATP
- ATPs can prove difficult theorems, just give them enough time
- a bit surprizingly, performance of ATPs on QPTP and UEQ TPTP is similar

Do you want your prover be used by mathematicians?

- Make it user friendly!
  - like CAS for calculus
  - or at least like Bill with Prover9/Mace4 GUI
  - care about output (we want to understand the proof!)
- Provide verifier
- Implement hints

Conclusions

- yes, we, mathematicians, want to use ATP
- ATPs can prove difficult theorems, just give them enough time
- a bit surprizingly, performance of ATPs on QPTP and UEQ TPTP is similar

Do you want your prover be used by mathematicians?

- Make it user friendly!
  - like CAS for calculus
  - or at least like Bill with Prover9/Mace4 GUI
  - care about output (we want to understand the proof!)
- Provide verifier
- Implement hints
- Implement hints without human interaction
- *Make it work within ZFC, or in HOL :-)*