Using Automated Theorem Provers in Non-associative Algebra

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

LPAR 2008, Doha, Qatar

[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 at ESARM

[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 at ESARM

This talk

- is about *solving open problems* by (first order) automated theorem provers
- is *not* about formal verification or theory formation, no toy examples

Areas of algebra

- simple axiomatization projects (about 10 papers, since early 90's)
- lattices with operators (about 10?)
  - Robbins problem
  - algebraic logic
- non-associative algebra
  - *quasigroups and loops* (about 25, since 1996)
  - etc. (about 5)

Areas of algebra

- simple axiomatization projects (about 10 papers, since early 90's)
- lattices with operators (about 10?)
  - Robbins problem
  - algebraic logic
- non-associative algebra
  - *quasigroups and loops* (about 25, since 1996)
  - etc. (about 5)

Types of computation

- direct proofs of difficult open problems
- proving tedious technical steps
- quickly checking easy conjectures
- exhaustive search

Main problems

1. formalization in FOL
   - almost nothing formalizable directly
   - sometimes a highly non-trivial task
   - which formalization is optimal
2. finding a proof
   - which prover, setting up parameters
3. reading and understanding the proof
   - yes, we want to understand it! (usually)
   - simplifying the proof
   - improving readability (introducing concepts, lemmas, etc.)

For every distribuive groupoid G, there exists a congruence α of G such that G/α is symmetric and all blocks of α are medial.

```
cnf(sos,axiom,mult(A,mult(B,C)) = mult(mult(A,B),mult(A,C))).
cnf(sos,axiom,mult(mult(A,B),C) = mult(mult(A,C),mult(B,C))).
```

```
cnf(goals,negate_conjecture,mult(mult(mult(a,b),mult(c,d)),
mult(mult(a,c),mult(b,d))) != mult(mult(mult(a,c),mult(b,d)),
mult(mult(a,b),mult(c,d)))).
```

```
cnf(goals,negated_conjecture,mult(mult(mult(a,b),mult(c,d)),
mult(mult(mult(a,b),mult(c,d)),mult(mult(a,c),mult(b,d)))) !=
mult(mult(a,c),mult(b,d))).
```

Bruck loops with abelian inner mapping group are centrally nilpotent.

```
cnf(sos,axiom,mult(unit,A) = A).
cnf(sos,axiom,mult(A,unit) = A).
cnf(sos,axiom,mult(A,i(A)) = unit).
cnf(sos,axiom,mult(i(A),A) = unit).
cnf(sos,axiom,i(mult(A,B)) = mult(i(A),i(B))).
cnf(sos,axiom,mult(i(A),mult(A,B)) = B).
cnf(sos,axiom,rd(mult(A,B),B) = A).
cnf(sos,axiom,mult(rd(A,B),B) = A).
cnf(sos,axiom,mult(mult(A,mult(B,A)),C) =
mult(A,mult(B,mult(A,C)))).
cnf(sos,axiom,mult(mult(A,B),C) =
mult(mult(A,mult(B,C)),asoc(A,B,C))).
cnf(sos,axiom,op_l(A,B,C) =
mult(i(mult(C,B)),mult(C,mult(B,A)))).
cnf(sos,axiom,op_r(A,B,C) = rd(mult(mult(A,B),C),mult(B,C))).
cnf(sos,axiom,op_t(A,B) = mult(i(B),mult(A,B))).
cnf(sos,axiom,op_r(op_r(A,B,C),D,E) = op_r(op_r(A,D,E),B,C)).
cnf(sos,axiom,op_l(op_r(A,B,C),D,E) = op_r(op_l(A,D,E),B,C)).
cnf(sos,axiom,op_l(op_l(A,B,C),D,E) = op_l(op_l(A,D,E),B,C)).
cnf(sos,axiom,op_t(op_r(A,B,C),D) = op_r(op_t(A,D),B,C)).
cnf(sos,axiom,op_t(op_l(A,B,C),D) = op_l(op_t(A,D),B,C)).
cnf(sos,axiom,op_t(op_t(A,B),C) = op_t(op_t(A,C),B)).

cnf(goals,negated_conjecture,asoc(asoc(a,b,c),d,e) != unit).
```

Our work, so far

- proving theorems
- QPTP library

QPTP = Quasigroup problems for theorem provers

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

- all papers covered, about 100 problems selected (about 80% equational)
- both formal (TPTP) and informal (paper) description
- downloadable at `www.karlin.mff.cuni.cz/~stanovsk/qptp`
- a benchmark (selected provers from CASC):

  Waldmeister >> E, Gandalf, Prover9, Vampire >> Spass

Summary

- (some) mathematicians use automated theorem provers
- ATPs can prove difficult theorems
- If you have a software that could solve our problems, let me know immediately!