Some Applications of Prover9: Status Report

Robert Veroff
Department of Computer Science
University of New Mexico

ADAM 2012
Northern Michigan University
June 12–14

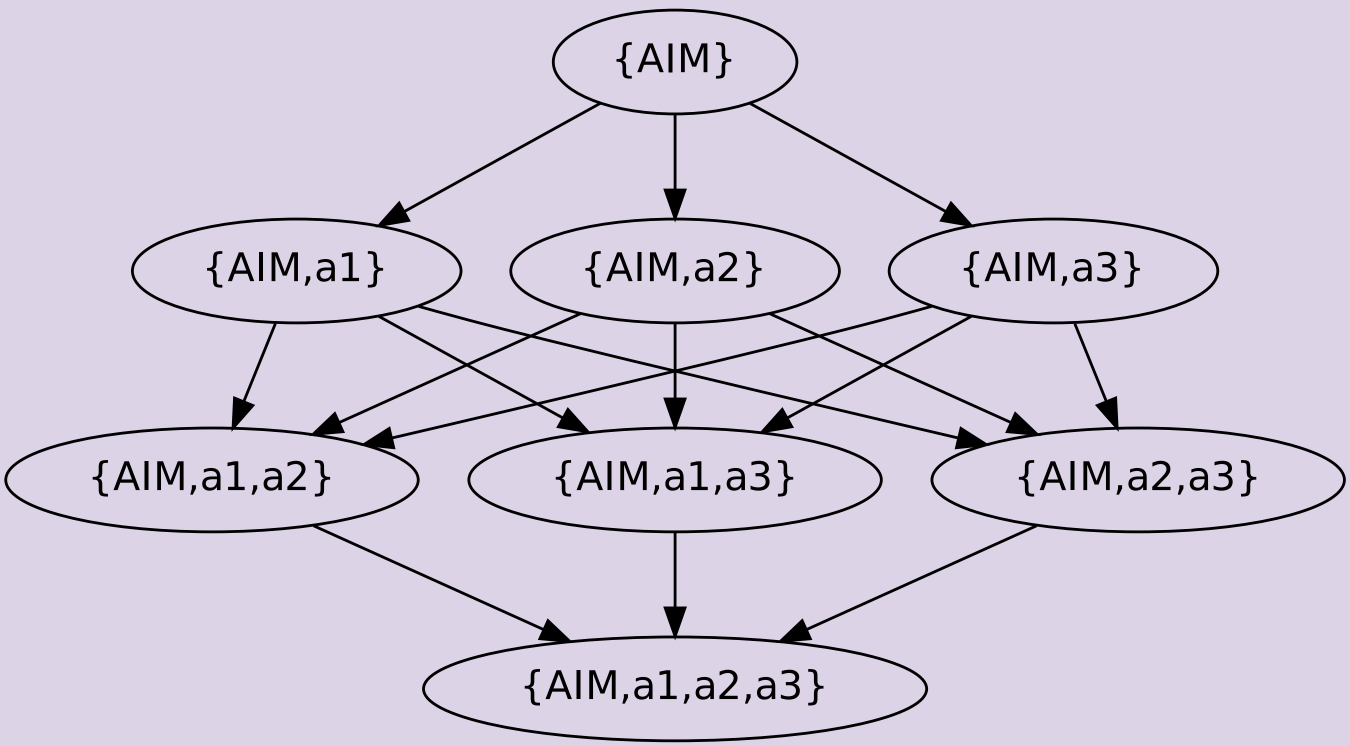
Application Areas

- Loop Theory (with Michael Kinyon, J. D. Phillips and Petr Vojtěchovský)
- Algebraic Geometry (with R. Padmanabhan)
- Algebraic Logic (with Matthew Spinks)

Loop Theory (AIM Problem)

- Concerns Abelian inner mappings
- Problem description
  - See file aim_descr.txt.
  - Several candidate extensions; some combinations of special interest
  - Working our way up the hierarchy

AIM Problem Hierarchy



Approach

Key observations:

- Proofs share many steps

  proof sketches – selection bias for proof steps of related theorems

- Sensitive to the lexical ordering of terms

  p9loop – iterate over multiple orderings, collecting hint matchers as lemmas for later iterations

Status

- Several of the cases of interest have been proved.
- The proofs tend to be *very* long by current AD standards.
  - evidence of the effectiveness of the methods
  - not black box solutions

Inference rule gL:

$$
\begin{array}{l}
(\forall \overrightarrow{x}, \overrightarrow{y}(\\
\quad (\exists z f(\overrightarrow{x}, z) = f(\overrightarrow{y}, z)) \\
\qquad \longrightarrow \\
\quad (\forall z f(\overrightarrow{x}, z) = f(\overrightarrow{y}, z)) \\
))
\end{array}
$$

Example gL rule:

$$
\begin{array}{l}
(\forall x_0, x_1, y_0, y_1(\\
\quad (\exists z (z * x_0) * x_1 = (z * y_0) * y_1)) \\
\qquad \longrightarrow \\
\quad (\forall z (z * x_0) * x_1 = (z * y_0) * y_1)) \\
))
\end{array}
$$

Corresponding clause:

```
(z * x0) * x1 != (z * y0) * y1
   | (w * x0) * x1 = (w * y0) * y1.
```

Inference Rule gL

gL clause:

```
(z * x0) * x1 != (z * y0) * y1
   | (w * x0) * x1 = (w * y0) * y1.
```

Example application of the rule:

```
(a * b) * c = (a * d) * e
```

resolves with the gL clause to produce

```
(w * b) * c = (w * d) * e
```

Automated Deduction Issues

Need a gL clause for every argument position!

- Nesting limit

  tradeoffs with Otter's built-in version

- Restrict application of gL clauses

  `set(para_units_only)`

For convenience, have a gL clause generator (Python script).

Algebraic Logic

Typical question: Let $\mathcal{S}$ and $\mathcal{T}$ be two algebras, deduction systems or logics. Under what extensions do $\mathcal{S}$ and $\mathcal{T}$ become definitionally equivalent?

The theorem proving tasks include proving various properties (in both directions).

$\mathcal{S} \cup e_{\mathcal{S}} \Rightarrow properties\ of\ \mathcal{T}$

$\mathcal{T} \cup e_{\mathcal{T}} \Rightarrow properties\ of\ \mathcal{S}$

The input sets are complex, often involving a large number of axioms, two sets of operations, for example, $\{\wedge, \vee, \neg, \sim, \rightarrow\}$ and $\{\wedge, \vee, *, \Rightarrow\}$, and definitions relating the operations of the two systems in question.

Standard methods (e.g., proof sketches) have not been all that effective by themselves. Progress has depended heavily on suggestions for intermediate results from Matthew.