Finite Model Search and Single Axioms

William McCune
Mathematics and Computer Science Division
Argonne National Laboratory

March 2002

Proofs and Countermodels

- Ordinary first-order theorem proving:
  - search for a proof of a conjecture,
  - show that a statement is unsatisfiable.
- We'll look at the other side of the question:
  - search for a counterexample to a conjecture,
  - show that a statement has a model.

Two Approaches to Equational Satisfiability

- Saturation with a complete inference system.
  - for example, unfailing Knuth-Bendix completion
  - if it terminates without a refutation, the statement is satisfiable
  - but we do not necessarily get a model
- Search for finite models (this is what we do).
  - iterate through domain sizes: 2, 3, 4, ...
  - each domain size is decidable by reduction to a ground problem
  - example programs: FINDER, MGTP, $\boxed{\text{MACE}}$, FALCON, STORK, $\boxed{\text{SEM}}$.
  - also, there are *local search* (incomplete) methods

Noncommutative 6-Group Example (MACE4)

Input

```
assign(domain_size, 6).
end_of_list.


e * x = x.
g(x) * x = e.
(x * y) * z = x * (y * z).

a * b != b * a.
```

Noncommutative 6-Group Example (MACE4)

Input

```
assign(domain_size, 6).
end_of_list.


e * x = x.
g(x) * x = e.
(x * y) * z = x * (y * z).

a * b != b * a.
```

Output

```
Model 1 at 0.00 seconds

* | 0 1 2 3 4 5
--+------------
0 | 0 1 2 3 4 5
1 | 1 0 4 5 2 3
2 | 2 3 0 1 5 4
3 | 3 2 5 4 0 1
4 | 4 5 1 0 3 2
5 | 5 4 3 2 1 0

g   0 1 2 3 4 5
---------------
    0 1 2 4 3 5

e:0   a:1   b:2
```

Two Methods (for the Finite Approach)

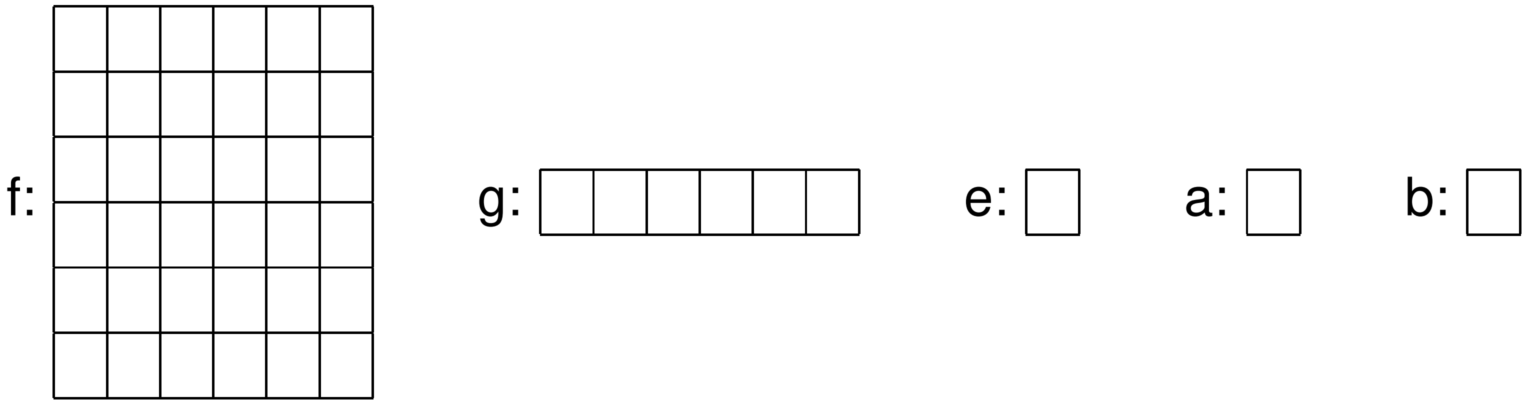
- Equational Method (SEM, MACE 4)
  - reduction to a ground problem with equality
  - decision by case splitting, and rewriting
  - (and possibly negative inference or some flattening)
- Relational (Flattening) Method (MACE 2)
  - reduction to a propositional problem (without equality)
  - decision by DPLL (Davis-Putnam-Loveland-Logeman)

Noncommutative 6-Group: Equational Method

Generate instances over a domain of $\{$0,1,2,3,4,5$\}$.

f(e,x)=x	f(g(x),x)=e	f(f(x,y),z)=f(x,f(y,z))	f(a,b)$\neq$f(b,a)
f(e,0)=0	f(g(0),0)=e	f(f(0,0),0)=f(0,f(0,0))	f(a,b)$\neq$f(b,a)
f(e,1)=1	f(g(1),1)=e	f(f(1,1),1)=f(1,f(1,1))	
f(e,2)=2	f(g(2),2)=e	f(f(1,0),1)=f(1,f(0,1))	
f(e,3)=3	f(g(3),3)=e	f(f(0,1),1)=f(0,f(1,1))	
f(e,4)=4	f(g(4),4)=e	f(f(0,0),1)=f(0,f(0,1))	
f(e,5)=5	f(g(5),5)=e	f(f(1,1),0)=f(1,f(1,0))	
		f(f(0,1),0)=f(0,f(1,0))	
		f(f(1,0),0)=f(1,f(0,0))	
		f(f(2,2),2)=f(2,f(2,2))	
		. . .	
		(207 more)	

Noncommutative 6-Group: Possible-Value Clauses



f(0,0)=0 $\vee$ f(0,0)=1 $\vee$ f(0,0)=2 $\vee$ f(0,0)=3 $\vee$ f(0,0)=4 $\vee$ f(0,0)=5.

$\cdots$ (35 more)

g(0)=0 $\vee$ g(0)=1 $\vee$ g(0)=2 $\vee$ g(0)=3 $\vee$ g(0)=4 $\vee$ g(0)=5.

$\cdots$ (5 more)

e=0 $\vee$ e=1 $\vee$ e=2 $\vee$ e=3 $\vee$ e=4 $\vee$ e=5.

a=0 $\vee$ a=1 $\vee$ a=2 $\vee$ a=3 $\vee$ a=4 $\vee$ a=5.

b=0 $\vee$ b=1 $\vee$ b=2 $\vee$ b=3 $\vee$ b=4 $\vee$ b=5.

Equational Decision Methods

- Given the instances of the input and the possible-value clauses,
  - Select a cell, say f(1,2).
  - for each possible value i of f(2,1),

    assign f(2,1)=i

    rewrite and propagate (this fills in other cells)

    if no contradiction, recurse and undo propagation
- How general should the rewriting be?
  - In MACE 4, we rewrite only with “assignment” equations, e.g., f(1,2)=3.
- Negative constraints

Noncommutative 6-Group: Relational (Flattening) Method

- Eliminate equality!
- Consider associativity of f: f(f(x,y),z) = f(x,f(y,z))
  - Transform it into a relational (flat) form:
    f(x,y)$\neq u$ $\vee$ f(y,z)$\neq v$ $\vee$ f(x,v)$\neq w$ $\vee$ f(u,z)=w.
  - Take instances over $\{$0,1,2,3,4,5$\}$. *We get* $6^6 = 46656$ *instances!*
- MACE does this kind of transformation for all clauses.
  - Add possible-value clauses (as in equational method).
  - Add clauses saying that each cell has at most one value.
  - Now we have a purely propositional problem. We use DPLL.

Relational (Flattening) Method

- Disadvantage: *memory requirements!*
  - Flattening $y * g(z * (((w * g(w)) * g(x * z)) * y)) = x$ produces 12 variables. For a domain of 6, that gives 2,176,782,336 instances.
- Advantage
  - Negative constraints are propagated as much as possible, automatically, by DPLL.
  - Negative constraints help, because of the possible-value clauses f(1,2)=0 $\vee$ f(1,2)=1 $\vee$ f(1,2)=2 $\vee$ f(1,2)=3 $\vee$ f(1,2)=4 $\vee$ f(1,2)=5.
- Intermediate method – equational with partial flattening.

Negative Constraints in the Relational Method

- MACE can make this inference propositionally:

$$\frac{\begin{array}{c} f(0,1) \neq 3 \;\vee\; f(1,2) \neq 4 \;\vee\; f(0,4) \neq 5 \;\vee\; f(3,2)=5 \\ f(0,1) = 3 \\ f(0,4) = 5 \\ f(3,2) \neq 5 \end{array}}{f(1,2) \neq 4}$$

- Without the transformation, we have to make a backward equational inference and a negative inference:

$$\frac{\begin{array}{c} f(f(0,1),2) = f(0,f(1,2)) \\ f(0,1) = 3 \end{array}}{f(3,2) = f(0,f(1,2))} \qquad \frac{\begin{array}{c} f(3,2) = f(0,f(1,2)) \\ f(3,2) \neq 5 \end{array}}{f(0,f(1,2)) \neq 5} \qquad \frac{\begin{array}{c} f(0,f(1,2)) \neq 5 \\ f(0,4) = 5 \end{array}}{f(1,2) \neq 4}$$

The Cells

This applies to both equational and relational methods

- How should the next cell be selected?
  - fewest possible values
  - smallest indices
  - most propagation (look ahead)
- Given a cell, which values need to be considered?
  - only the “possible values”
  - consider symmetry of elements not yet constrained

Isomorphism Reduction

- Consider the search for noncommutative groups of order 6.
  - We have 3 constants: $\{$e,a,b$\}$.
  - Without loss of generality, they can be distinct, so we can assign them to members of the domain, say $\{$0,1,2$\}$.
  - This reduces the number of models from 2160 to 18.
- Least Number Heuristic (SEM)
  - Use a “smallest index” cell selection strategy, and assign the smallest values first.
  - If the smallest unconstrained value is n, consider elements $\{0, \cdots, n\}$.
  - In the NC 6-Group example, this reduces the number of models to 3.

Applications to Quantum Logic

Lattice Axioms

$$\begin{array}{ll} x \wedge y = y \wedge x & x \vee y = y \vee x \\ (x \wedge y) \wedge z = x \wedge (y \wedge z) & (x \vee y) \vee z = x \vee (y \vee z) \\ (x \vee y) \wedge x = x & (x \wedge y) \vee x = x \end{array}$$

Add these to get an Ortholattice

$$\begin{array}{ll} x \vee c(x) = 1 & x \wedge c(x) = 0 \\ c(c(x)) = x & c(x \vee y) = c(x) \wedge c(y) \end{array}$$

Weak Orthomodular law

$$(c(x) \wedge (x \vee y)) \vee (c(y) \vee (x \wedge y)) = 1$$

Does the following hold in all weakly orthomodular lattices? [Norm Megill].

$$x \wedge (y \vee (x \wedge (c(x) \vee (x \wedge y)))) = x \wedge (c(x) \vee (x \wedge y)) \qquad \text{(*3-M68)}$$

Quantum Logic

Model 1 at 0.87 seconds

^	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	0	2	2	6	0	0	6	10	12	13	10	0	12	13	10	19	13	13	0	19
3	0	3	6	3	0	0	6	11	6	6	0	11	6	6	11	11	3	6	0	0
4	0	4	0	0	4	0	0	4	4	0	0	0	0	0	4	4	0	4	0	0
5	0	5	0	0	0	5	0	5	0	5	0	0	0	0	18	18	0	0	18	0
6	0	6	6	6	0	0	6	0	6	6	0	0	6	6	0	0	6	6	0	0
7	0	7	10	11	4	5	0	7	4	5	10	11	0	0	14	15	11	4	18	19
8	0	8	12	6	4	0	6	4	8	12	0	0	12	12	4	4	12	8	0	0
9	0	9	13	6	0	5	6	5	12	9	0	0	12	13	18	18	13	13	18	0
10	0	10	10	0	0	0	0	10	0	0	10	0	0	0	10	19	0	0	0	19
11	0	11	0	11	0	0	0	11	0	0	0	11	0	0	11	11	11	0	0	0
12	0	12	12	6	0	0	6	0	12	12	0	0	12	12	0	0	12	12	0	0
13	0	13	13	6	0	0	6	0	12	13	0	0	12	13	0	0	13	13	0	0
14	0	14	10	11	4	18	0	14	4	18	10	11	0	0	14	15	11	4	18	19
15	0	15	19	11	4	18	0	15	4	18	19	11	0	0	15	15	11	4	18	19
16	0	16	13	3	0	0	6	11	12	13	0	11	12	13	11	11	16	13	0	0
17	0	17	13	6	4	0	6	4	8	13	0	0	12	13	4	4	13	17	0	0
18	0	18	0	0	0	18	0	18	0	18	0	0	0	0	18	18	0	0	18	0
19	0	19	19	0	0	0	0	19	0	0	19	0	0	0	19	19	0	0	0	19

v	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	2	1	1	1	2	1	1	1	2	1	2	2	1	1	1	1	1	2
3	3	1	1	3	1	1	3	1	1	1	1	3	16	16	1	1	16	1	1	1
4	4	1	1	1	4	7	8	7	8	1	14	15	8	17	14	15	1	17	15	15
5	5	1	1	1	7	5	9	7	1	9	7	7	9	9	7	7	1	1	5	7
6	6	1	2	3	8	9	6	1	8	9	2	3	12	13	1	1	16	17	9	2
7	7	1	1	1	7	7	1	7	1	1	7	7	1	1	7	7	1	1	7	7
8	8	1	1	1	8	1	8	1	8	1	1	1	8	17	1	1	1	17	1	1
9	9	1	1	1	1	9	9	1	1	9	1	1	9	9	1	1	1	1	9	1
10	10	1	2	1	14	7	2	7	1	1	10	14	2	2	14	14	1	1	14	10
11	11	1	1	3	15	7	3	7	1	1	14	11	16	16	14	15	16	1	15	15
12	12	1	2	16	8	9	12	1	8	9	2	16	12	13	1	1	16	17	9	2
13	13	1	2	16	17	9	13	1	17	9	2	16	13	13	1	1	16	17	9	2
14	14	1	1	1	14	7	1	7	1	1	14	14	1	1	14	14	1	1	14	14
15	15	1	1	1	15	7	1	7	1	1	14	15	1	1	14	15	1	1	15	15
16	16	1	1	16	1	1	16	1	1	1	1	16	16	16	1	1	16	1	1	1
17	17	1	1	1	17	1	17	1	17	1	1	1	17	17	1	1	1	17	1	1
18	18	1	1	1	15	5	9	7	1	9	14	15	9	9	14	15	1	1	18	15
19	19	1	2	1	15	7	2	7	1	1	10	15	2	2	14	15	1	1	15	19

c	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	1	0	4	5	2	3	7	6	10	11	8	9	14	15	12	13	18	19	16	17

A = 2 B = 3

A Weakly Orthomodular Lattice Violating (*3-M68)

Quantum Logic

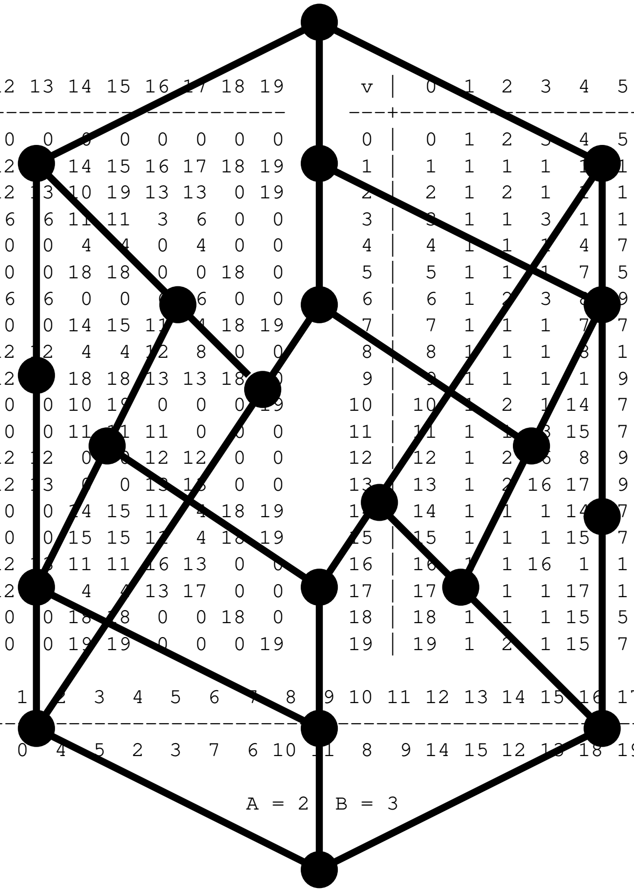
Model 1 at 0.87 seconds

^	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	[illegible]	14	15	16	17	18	19
2	0	2	2	6	0	0	6	10	12	13	10	0	12	13	10	19	13	13	0	19
3	0	3	6	3	0	0	6	11	6	6	0	11	6	6	11	11	3	6	0	0
4	0	4	0	0	4	0	0	4	4	0	0	0	0	0	4	4	0	4	0	0
5	0	5	0	0	0	5	0	5	0	5	0	0	0	0	18	18	0	0	18	0
6	0	6	6	6	0	0	6	0	6	6	0	0	6	6	0	0	[illegible]	6	0	0
7	0	7	10	11	4	5	0	7	4	5	10	11	0	0	14	15	11	4	18	19
8	0	8	12	6	4	0	6	4	8	12	0	0	12	12	4	4	12	8	0	0
9	0	9	13	6	0	5	6	5	12	9	0	0	12	[illegible]	18	18	13	13	18	0
10	0	10	10	0	0	0	0	10	0	0	10	0	0	0	10	19	0	0	0	19
11	0	11	0	11	0	0	0	11	0	0	0	11	0	0	11	11	11	0	0	0
12	0	12	12	6	0	0	6	0	12	12	0	0	12	12	0	[illegible]	12	12	0	0
13	0	13	13	6	0	0	6	0	12	13	0	0	12	13	0	0	13	13	0	0
14	0	14	10	11	4	18	0	14	4	18	10	11	0	0	14	15	11	4	18	19
15	0	15	19	11	4	18	0	15	4	18	19	11	0	0	15	15	11	4	18	19
16	0	16	13	3	0	0	6	11	12	13	0	11	12	13	11	11	16	13	0	0
17	0	17	13	6	4	0	6	4	8	13	0	0	12	[illegible]	4	4	13	17	0	0
18	0	18	0	0	0	18	0	18	0	18	0	0	0	0	18	18	0	0	18	0
19	0	19	19	0	0	0	0	19	0	0	19	0	0	0	19	19	0	0	0	19

v	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	2	1	1	1	2	1	1	1	2	1	2	2	1	1	1	1	1	2
3	3	1	1	3	1	1	3	1	1	1	1	3	16	16	1	1	16	1	1	1
4	4	1	1	1	4	7	8	7	8	1	14	15	8	17	14	15	1	17	15	15
5	5	1	1	1	7	5	9	7	1	9	7	7	9	9	7	7	1	1	5	7
6	6	1	2	3	8	9	6	1	8	9	2	3	12	13	1	1	16	17	9	2
7	7	1	1	1	7	7	1	7	1	1	7	7	1	1	7	7	1	1	7	7
8	8	1	1	1	8	1	8	1	8	1	1	1	8	17	1	1	1	17	1	1
9	9	1	1	1	1	9	9	1	1	9	1	1	9	9	1	1	1	1	9	1
10	10	1	2	1	14	7	2	7	1	1	10	14	2	2	14	14	1	1	14	10
11	11	1	1	3	15	7	3	7	1	1	14	11	16	16	14	15	16	1	15	15
12	12	1	2	6	8	9	12	1	8	9	2	16	12	13	1	1	16	17	9	2
13	13	1	2	16	17	9	13	1	17	9	2	16	13	13	1	1	16	17	9	2
14	14	1	1	1	14	7	1	7	1	1	14	14	1	1	14	14	1	1	14	14
15	15	1	1	1	15	7	1	7	1	1	14	15	1	1	14	15	1	1	15	15
16	16	1	1	16	1	1	16	1	1	1	1	16	16	16	1	1	16	1	1	1
17	17	[illegible]	1	1	17	1	17	1	17	1	1	1	17	17	1	1	1	17	1	1
18	18	1	1	1	15	5	9	7	1	9	14	15	9	9	14	15	1	1	18	15
19	19	1	2	1	15	7	2	7	1	1	10	15	2	2	14	15	1	1	15	19

c	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	1	0	4	5	2	3	7	6	10	11	8	9	14	15	12	13	18	19	16	17

A = 2 B = 3



A Weakly Orthomodular Lattice Violating (*3-M68)

The Search for Short Single Axioms

- Is this a single axiom for Boolean Algebra in terms of OR and NOT?

$$\neg(\neg(x \vee \neg(\neg(y \vee x) \vee x)) \vee \neg(y \vee \neg(z \vee x))) = y.$$

The Search for Short Single Axioms

- Is this a single axiom for Boolean Algebra in terms of OR and NOT?

$$\neg(\neg(x \vee \neg(\neg(y \vee x) \vee x)) \vee \neg(y \vee \neg(z \vee x))) = y.$$

- No — a counter model in which $\exists A (A \vee A) \neq A$.

```
or: | 0 1 2 3      not:  0 1 2 3      A: 0
  --+--------        -----------
  0 | 1 0 2 3            0 1 3 2
  1 | 0 0 0 0
  2 | 2 0 2 0
  3 | 3 0 0 3
```

Single Axiom Search Methods

Consider lattice theory

- Enumerate well-formed equations.
- Use a decision procedure to remove non-identities.
- Maintain a set of non-lattices (found by the model searcher).
- Remove candidates true in any of the non-lattices.
- Use model searcher to look for a non-lattice model.
  - If found, add it to the set of non-lattices;
  - otherwise, use a theorem prover (Otter) to search for a proof that the candidate is a single axiom.

New Single Axioms

- Boolean Algebra (disjunction and negation)

$$\neg(\neg(x \vee y) \vee \neg z) \vee \neg(\neg(\neg u \vee u) \vee (\neg z \vee x)) = z$$

Length 22. The shortest previously known axiom has length 131.

- Lattice Theory (meet and join)

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

Length 29. The shortest previously known axiom has length 79.

Programs

- MACE 2 (old MACE)
  - Reduction to a purely propositional problem (by flattening).
  - Good performance if formulas are not complicated (flat).
  - Language (and input files) same as for our theorem prover Otter.
  - `www.mcs.anl.gov/AR/mace`
- SEM
  - Reduction to a ground problem with equality.
  - Very good performance for equational problems.
  - Can handle sorted logic; that is, several disjoint domains.
  - `www.cs.uiowa.edu/~hzhang/sem.html`
- MACE 4 (under construction)
  - Reduction to a ground problem with equality.
  - With some flattening.

Conclusions

- Finite model search works!
  - *new results*
  - independence of axiom systems
  - exploring theories
  - as Otter's companion
- There should be important applications in other areas.
  - design problems
  - systems verification – model checking?
- The future looks good for more powerful methods.