Automated Deduction

William McCune and Larry Wos

Mathematics and Computer Science Division
Argonne National Laboratory

October 16, 1998

The Zebra Puzzle

There are five houses in a row. Each is associated with a color, a nationality, a pet, a drink, and a type of cigarette.

The englishman lives in the red house, the Spaniard has a dog, coffee is drunk in the green house, the Ukranian drinks tea, the person who smokes Old Golds has a snail, the person who smokes Kools lives in the yellow house, milk is drunk in the third house, the Norwegian lives in the first house, the person who smokes Lucky Strikes lives in the orange house, the Japanese smokes Parliaments, and the second house is blue.

The ivory house is just to the left of the green house, Kools are smoked next door to the horse's house, and Chesterfields are smoked next to the fox's house.

Is there a solution?

Schubert's Steamroller

1. Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants.

2. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants.

3. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which are in turn much smaller than wolves.

4. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Is there is an animal that likes to eat an animal that eats all grains?

A formalization of statement 2:

$$\forall x \left(animal(x) \rightarrow \left[\begin{array}{l} \forall y (plant(y) \rightarrow eats(x,y)) \vee \\ \forall z \left[\begin{array}{l} animal(z) \ \& \\ smaller(z,x) \ \& \\ \exists u \left[\begin{array}{l} plant(u) \ \& \\ eats(z,u) \end{array} \right] \end{array} \right] \rightarrow eats(x,z) \end{array} \right] \right)$$

Automated Deduction

- Goal: deductive support for mathematicians, scientists, engineers, and others.
- Languages/Logics
  - propositional, *first-order*, higher order
  - *classical* vs. constructive
- Methods
  - *resolution, paramodulation, rewriting, with strategy to control search*
  - tableau
  - *automatic* — — — — interactive
  - AI
- Current Applications
  - *mathematics and logic*
  - *computer systems design and verification*

Programs We Use

- Developed here at MCS
  - Otter: a theorem prover for first-order logic with equality
  - EQP: a theorem prover for equational logic with associative-commutative unification
  - MACE: a program to search for counterexamples and other kinds of models
- ACL2 (UT Austin): a program for proving properties of a subset of Common Lisp.

Logics for Quantum Computing

Quantum logic models probability of events in quantum dynamics, and may serve as a basis for computing with quantum devices.

An *Ortholattice* (OL) satisfies

$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
$(x \vee y) \vee z = x \vee (y \vee z)$	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$
$x'' = x$	
$(x \wedge y)' = x' \vee y'$	
$x \vee x' = 1$	$x \wedge x' = 0$
$x \vee 1 = 1$	$x \wedge 0 = 0$

(like Boolean algebra, but not necessarily distributive)

An *orthomodular lattice* (OML) also satisfies

$$x \vee (x' \wedge (x \vee y)) = x \vee y$$

Ortholattice Questions

(from Norman Megill and Mladen Pavičić)

These questions arose when analyzing implication rules for quantum logics.

The following three equations hold for OML. Do they hold also for OL?

$$((a \wedge b') \vee a')' \vee ((a \wedge b') \vee ((a' \wedge ((a \vee b') \wedge (a \vee b))) \vee (a' \wedge ((a \vee b') \wedge (a \vee b))')))) = 1 \qquad \text{(E1)}$$

$$(a \vee ((a' \wedge ((a \vee b') \wedge (a \vee b))) \vee (a' \wedge ((a' \wedge b) \vee (a' \wedge b'))))) = 1 \qquad \text{(E2)}$$

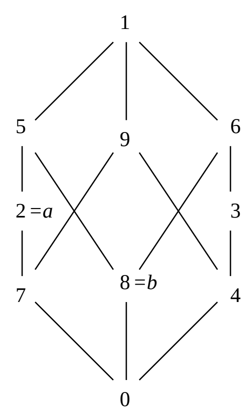
$$(((a' \wedge b) \vee (a' \wedge b')) \vee (a \wedge (a' \vee b)))' \vee (a' \vee b) = 1 \qquad \text{(E3)}$$

The following *fails* for OL.
Does it hold for OML?

$$a' \vee ((b' \vee a) \wedge (b \vee (b' \wedge a))) = 1 \qquad \text{(E7)}$$

Ortholattice Answers

- E1. MACE found a counterexample of size 10 (in 20 minutes):



An ortholattice violating E1.

- OL $\Rightarrow$ E2 (in 4 seconds).
- OL $\Rightarrow$ E3 (in 23 hours).
- OML $\Rightarrow$ E7 (in 13 minutes).

Deductive Program Synthesis Example

Amphion – SRI, NASA Ames, JPL (1994)

- SPICELIB: a FORTRAN library for solar system computations
- Axioms: SPICELIB specifications (and a few other axioms about programs)
- Say you need a program P that implements specification S (a GUI helps you construct S).
- Conjecture: $\exists$ a program P that satisfies S.
- Prove it (automatically).
- Extract program P (automatically) from the proof.
- Program P implements specification S.

Are the Program's Proofs Correct?

```
2  x+y=y+x.
3,4  (x+y)+z=x+y+z.
5  n(n(n(x)+y)+n(x+y))=y.
6  n(x+y)!=n(x).
64  x+y+z=y+x+z.  [para_into,4,2,demod,3]
71  n(n(n(x)+y)+n(y+x))=y.  [para_into,5,2]
73  n(n(x+y)+n(n(x)+y))=y.  [para_into,5,2]
75  n(x+y)!=n(y).  [para_into,6,2]
93  n(n(x+y)+n(n(y)+x))=x.  [para_into,71,2]
116  n(x+y+z)!=n(z).  [para_into,75,4]
130  n(n(n(n(x)+y)+x+y)+y)=n(n(x)+y).  [para_into,93,73]
132  n(n(n(x+y)+n(x)+y)+y)=n(x+y).  [para_into,93,5]
139  n(x+y+z+u)!=n(y+u).  [para_into,116,64]
170  n(n(n(n(x)+y)+x+y+y)+n(n(x)+y))=y.  [para_from,130,73,demod,3,3]
174  n(n(n(x)+y)+n(n(n(x)+y)+x+y+y))=y.  [para_from,130,5,demod,3,3]
190  n(n(n(x+y+z)+y+n(x)+z)+y+z)=n(x+y+z).  [para_into,132,64]
211  n(x+y+z+u)!=n(x+u).  [para_into,139,64]
253  n(n(n(n(x)+y)+n(n(x)+y)+x+y+y)+y)=n(n(x)+y).  [para_from,170,93]
260  n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+z)+n(y+z))=z.  [para_from,170,73,demod,3]
346  n(n(n(x+y+z)+y+n(x)+z)+z+y)=n(x+y+z).  [para_into,190,2]
423  n(n(n(x)+y)+n(n(n(x)+y)+n(n(x)+y)+x+y+y+y))=y.  [para_from,253,5,demod,3,3,3,3]
475  n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+z+n(y+z))+z)=n(y+z).  [para_from,260,73,demod,3,3]
486  n(n(n(x+y+z)+x+n(z)+y)+y+x)=n(z+x+y).  [para_into,346,2,demod,3]
684  n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+n(y+z)+z)+z)=n(y+z).  [para_into,475,2]
801  n(n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+n(y+z)+z)+z+u)+n(n(y+z)+u))=u.  [para_from,684,73,demod,3]
848  n(n(x+x+n(n(x)+x)+x)+n(n(x+x)+n(n(x)+x)))=n(n(x)+x).  [para_into,801,486,demod,3]
878  n(n(x+x+n(n(x)+x)+x)+x)=n(n(x)+x).  [para_into,848,93]
897  n(n(x+n(n(x)+x)+x+x)+x)=n(n(x)+x).  [para_into,878,64]
948  n(n(n(n(x)+x)+x+x+x)+x)=n(n(x)+x).  [para_into,897,64]
1015  n(n(n(n(n(x)+x)+x+x+x)+x+y)+n(n(n(x)+x)+y))=y.  [para_from,948,73,demod,3]
1036  n(n(n(n(x)+x)+x+x+x+x)+n(n(x)+x))=x.  [para_from,948,73,demod,3,3,3]
1053  n(n(n(n(n(x)+x)+x+x+x)+x+n(n(n(x)+x)+n(n(x)+x)+x+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x).  [para_into,1015,423]
1079  n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x)=n(n(x)+x).  [para_from,1036,93]
1112  n(n(x+n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x).  [para_into,1053,2,demod,3]
1130  n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+y)+n(n(n(x)+x)+y))=y.  [para_from,1079,73,demod,3]
1149  n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x).  [para_into,1112,64]
1169  n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+x+x+x).  [para_into,1130,174]
1211  n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)=n(n(n(x)+x)+x+x+x).  [para_into,1169,1149]
1212  $F.  [binary,1211,211]
```


Program Verification with ACL2

Consider sorting a list.

Definitions:

$permutation(A, B)$: List A is a permutation of list B.
$ordered(A)$: The members of list A are nondecreasing.

Program:

$sort(A)$: A program that sorts list A and returns the result, using your favorite sorting algorithm.

Theorem to Prove (specification):

$$\forall A \left(\begin{array}{c} ordered(sort(A)) \ \& \\ permutation(sort(A), A) \end{array} \right)$$

Applications of ACL2 and its Predecessor

- CLInc Stack (Boyer, Moore, Hunt, et al, UT Austin)
  - Hardware design: down to the gate level
  - Software: assembler, compiler, simple operating system
- AMD K5, K7 processor floating point design (Russinoff, AMD)
- Verified theorem prover (here at MCS)