Equational Deduction and Applications to Boolean Algebra, Quantum Logic, and Cubic Curves

William McCune
mccune@mcs.anl.gov
www.mcs.anl.gov/~mccune

Mathematics and Computer Science Division
Argonne National Laboratory
U.S.A.

presented at
Mechanized Mathematics Research Center
Academia Sinica
Beijing

December 2000

Knuth-Bendix Completion

A minimal basis for group theory

$$e * x = x$$
$$x^{-1} * x = e$$
$$(x * y) * z = x * (y * z)$$

A *good* set of rules for simplifying group theory expressions

$$\begin{aligned} e * x &\longrightarrow x \\ x * e &\longrightarrow x \\ x^{-1} * x &\longrightarrow e \\ x * x^{-1} &\longrightarrow e \\ e^{-1} &\longrightarrow e \\ (x^{-1})^{-1} &\longrightarrow x \\ (x * y)^{-1} &\longrightarrow y^{-1} * x^{-1} \\ (x * y) * z &\longrightarrow x * (y * z) \\ x * (x^{-1} * y) &\longrightarrow y \\ x^{-1} * (x * y) &\longrightarrow y \end{aligned}$$

The Knuth-Bendix Procedure

- Infer new equations (*superposition*).
- Try to make all equations into (terminating) rewrite rules.
- Keep all equations reduced with respect to each other.

If the procedure succeeds, we have a

complete set of reductions,

which is a decision procedure for equality of terms in the equational theory.

Failure of Knuth-Bendix Completion

- A complete set of reductions might not exist.
  - undecidablility of the word problem
- It might exist, but be hard to find.
  - a bad term ordering
  - hard to prove termination
  - take a long time
- What if a derived equation cannot be made into a rewrite rule?
  - For example, as a rewrite rule,

    $$x * y = y * x$$

    does not terminate.

Extensions of K-B Completion

- Build equations into the inference and simplification processes.
  - commutativity (C)
  - associativity and commutativity (AC)
  - arbitrary equational theory (E-unification)
- Unfailing completion
  - Use non-orientable equations as constrained rewrite rules.
- Non-equational theories
- Search for proofs
  - Deny the goal; search for a contradiction.
  - Don't worry about non-orientable equations.
  - This can be a complete proof procedure for equational logic.

How do We Search for Proofs?

- Start with a procedure similar to Knuth-Bendix completion.
- For theories with associative-commutative (AC) operators, use AC unification instead of AC equations.
  - *with a heuristic*
- Don't worry about completeness.
- Use *basic paramodulation*, a refinement (constraint) of the inference process.
- Select *carefully* the inferences to make.
  - combine breadth-first and shortest-first search
- If the search fails, examine the output, and try again with a different search strategy.

Otter Theorem Prover

- Developed at Argonne National Laboratory, U.S.A, starting in 1988.
- Theorem prover for first-order logic with equality.
- Inference rules based on resolution and paramodulation.
- Operates on clauses, including equations.
  - general formulas are translated to clauses
- For this work, we used the *equational* capabilities of Otter.
  - search based on unfailing Knuth-Bendix completion
  - simplification

How Otter Works

- Not interactive—the user gives a conjecture and a search strategy in an input file.
- All proofs are by contradiction.
- The search is driven by a simple loop.
  - apply inference rules selected by the user
  - *carefully select clauses for inference*
  - simplify new equalities, test for subsumption
  - use new equalities to simplify old equalities
  - apply other search strategies
  - stop when a contradiction is found or when resources are exhausted.
- The user frequently makes a sequence of runs, examining the output and adjusting the search parameters.

Achieving Well Behaved Search

- Limit size of equations.
  - Incomplete, but necessary in practice.
  - Find a *good* limit.
- Compare results of different trial searches.
  - If a search is more economical and it gives a similar set of equations, use it.
- Restart the search with new lemmas.
  - Clean out the deadwood and focus on the lemmas.
- Observe derived equations, try to get more syntactic variety, iterate.

Weaknesses of Otter

- Limited to first-order problems.
  - *this is also a strength!*
- Theories with many operations.
- Defined operations.
- Non-Horn theories, set theory (Otter).

Set theory has all of these properties!

- Most proofs are longer than necessary.
- You have to learn a lot about the programs before you can use them for difficult problems.
- Associative-commutative operations.

Associative-Commutative Unification

- Assume "$+$" is AC.
- AC-identical terms:

$$(B + x) + (y + A), \qquad x + (y + (A + B))$$

- AC unification: making two terms AC-identical by instanciating variables.
- Ex: $(x + y + A)$, $(u + v + B)$ [A, B constants]

x	y	u	v
B	v	A	v
v	B	A	v
$u + B$	y	u	$y + A$
$z + u + B$	y	u	$z + y + A$

and so on, for a total of 40 AC unifiers.

- Worst case: doubly exponential number of AC unifiers.

Heuristic for AC Unification

- The pair of terms $(x+x+x)$ and $(y+z+u+v)$ has 1,044,569 most general AC unifiers.
- A heuristic:
  - Use only the simplest AC unifiers.
  - This is an incomplete (and severe) restriction.
  - Applied to the pair above, we get 139 AC unifiers.
  - In practice, it does not block many proofs.

The Program EQP

- Developed at Argonne, starting in 1993.
- Many similarities to Otter.
- Major differences from Otter:
  - restricted to equational logic
  - AC unification for inference
  - AC matching for simplification
- In practice, it is useful for problems about lattice structures.

Model Searching

- Search for finite models of first-order statements.
- Example, search for a non-commutative group. MACE Input:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
op(400, xfx, *).  op(300, yf, ^).
list(usable).

e * x = x.                        % Group Axioms
x^ * x = e.
(x * y) * z = x * (y * z).

A * B != B * A.      % A and B do not commute
end_of_list.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


MACE Output for $n = 6$

```
====================== Model #1 at 0.33 seconds:

 * :                       ^ :
        | 0 1 2 3 4 5           0 1 2 3 4 5
      --+------------        ---------------
      0 | 0 1 2 3 4 5           0 1 2 4 3 5
      1 | 1 0 3 2 5 4
      2 | 2 4 0 5 1 3
      3 | 3 5 1 4 0 2      e : 0
      4 | 4 2 5 0 3 1      A : 1
      5 | 5 3 4 1 2 0      B : 2

end_of_model
```

How can we use these programs?

- Search for counterexamples in parallel with search for proof.
- If we seek a single axiom for Abelian groups, we can eliminate any candidate for which this structure is a model.

Model-Searching Programs

- Programs
  - FINDER (Slaney, 1990)
  - MACE (McCune, 1992)
  - SEM (Jian Zhang, 1995)
- For most interesting theories, these programs can find only small models, with less than 10–20 elements.
- SEM is usually better than MACE or FINDER.

Applications of Otter, EQP, MACE

- Quantum Logic–orthomodular lattices
- Boolean Algebra
  - Robbins problem
  - single axioms
- Algebraic Geometry—cubic curves

Confession: *When applying automated deduction, I usually do not learn much about the mathematics of the application.*

Quantum Logic

An *Ortholattice* (OL) satisfies

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

Almost a Boolean algebra, but not distributive.

An *orthomodular lattice* (OML) also satisfies

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

- Ordinary probability: Boolean algebra over a set of events.
- Probability in quantum dynamics: Orthomodular lattice over a Hilbert space.

Ortholattice Questions

(from Norman Megill)

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

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

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

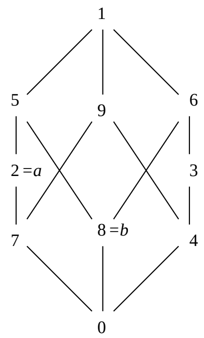
$$(((a' \wedge b) \vee (a' \wedge b')) \vee (a \wedge (a' \vee b)))' \vee (a' \vee b) = 1 \qquad \text{(E3)}$$

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: counterexample (by MACE, in 20 min.):



An ortholattice violating E1.

- E2: proof (by EQP, in 4 sec.), OL $\Rightarrow$ E2.
- E3: proof (by EQP, in 23 hr.), OL $\Rightarrow$ E3.
- E7: proof (by EQP, in 13 min.), OML $\Rightarrow$ E7.

The Robbins Problem

- A well-known basis for Boolean algebra [Huntington, 1933]:

$$\begin{array}{ll} x + y = y + x & \text{(C) Commutativity} \\ (x+y)+z = x+(y+z) & \text{(A) Associativity} \\ \overline{(\overline{x}+y)} + \overline{(\overline{x}+\overline{y})} = x & \text{(H) Huntington} \end{array}$$

- The Robbins Problem:

$$\overline{(\overline{(x+\overline{y})} + \overline{(x+y)})} = x \quad \text{(R) Robbins}$$

  Is $\{C, A, R\}$ basis for Boolean algebra?

- A proof of the following would solve the problem.

$$\text{Robbins} \Longrightarrow_{\text{AC}} \text{Huntington}$$

Easy Lemmas About the Robbins Equation

A very easy one:

$$\left\{ \begin{array}{l} \text{Robbins} \\ \forall x\ (\overline{\overline{x}} = x) \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Another easy one (but not first-order):

$$\left\{ \begin{array}{l} \text{Robbins} \\ \text{over a finite set} \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Existence of a zero:

$$\left\{ \begin{array}{l} \text{Robbins} \\ \exists x \forall y\ (x + y = y) \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Idempotence:

$$\left\{ \begin{array}{l} \text{Robbins} \\ \forall x\ (x + x = x) \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Existence of an idempotent element:

$$\left\{ \begin{array}{l} \text{Robbins} \\ \exists x\ (x + x = x) \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Hard Lemmas About the Robbins Equation

These were formulated and proved by Steve Winker c. 1980 [J. Algebra 1992].

Lemma 1 (An absorption):

$$\left\{ \begin{array}{l} \text{Robbins} \\ \exists x \exists y\ (x + y = x) \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Lemma 2 (An absorption within complement):

$$\left\{ \begin{array}{l} \text{Robbins} \\ \exists x \exists y\ (\overline{x + y} = \overline{x}) \end{array} \right\} \Longrightarrow_{\text{AC}} \text{Huntington.}$$

Winker's proofs are not first-order (induction).

These lemmas were proved automatically with EQP in 1992 and 1995.

The Final Assault

- About 50 searches with various strategies, including 14 multiday searches.
  - Many small attempts to achieve *well-behaved search*.
- Success: Robbins $\Longrightarrow_{\text{AC}} \exists x \exists y\ (\overline{x+y} = \overline{x})$
  - 8 day search, 30 megabytes of memory.
  - Basic paramod., AC heuristic.
  - Small inference tasks.
  - Equal parts breadth-first, shortest-first search.
  - Discard equations with more than 70 symbols.
- 5 weeks of real time, 5 weeks of CPU time.
- Quicker proofs and shorter proofs were found later.

Solutions of the Robbins Problem

First proof (EQP, 8 days of search):

$$\text{Robbins} \Longrightarrow_{\mathrm{AC}} \exists x \exists y\ (\overline{x+y}=\overline{x})$$

Another proof (EQP, a bit more search):

$$\text{Robbins} \Longrightarrow_{\mathrm{AC}} \exists x \exists y\ (x+y=x)$$

Yet another proof (EQP, 10%–90% more search):

$$\text{Robbins} \Longrightarrow_{\mathrm{AC}} \text{Huntington}$$

A Proof that Solves the Robbins Problem

Theorem. *Robbins* $\Longrightarrow_{\mathrm{AC}} \exists c \exists d\ (c+d=c)$.

7	$\overline{\overline{\overline{x}+y}+\overline{x+y}}=y$	[Robbins Equation]
10	$\overline{\overline{\overline{x+y}+\overline{x}+y}+y}=\overline{x+y}$	[7 → 7]
11	$\overline{\overline{\overline{\overline{x}+y}+x+y}+y}=\overline{\overline{x}+y}$	[7 → 7]
29	$\overline{\overline{\overline{\overline{x}+y}+x+2y}+\overline{\overline{x}+y}}=y$	[11 → 7]
54	$\overline{\overline{\overline{\overline{\overline{x}+y}+x+2y}+\overline{\overline{x}+y}+z}+\overline{y+z}}=z$	[29 → 7]
217	$\overline{\overline{\overline{\overline{\overline{x}+y}+x+2y}+\overline{\overline{x}+y}+\overline{y+z}+z}+z}=\overline{y+z}$	[54 → 7]
674	$\overline{\overline{\overline{\overline{\overline{\overline{x}+y}+x+2y}+\overline{\overline{x}+y}+\overline{y+z}+z}+z+u}+\overline{\overline{y+z}+u}}=u$	[217 → 7]
6736	$\overline{\overline{\overline{\overline{3x}+x}+\overline{3x}}+\overline{\overline{\overline{3x}+x}+5x}}=\overline{\overline{3x}+x}$	[10 → 674]
8855	$\overline{\overline{\overline{3x}+x}+5x}=\overline{3x}$	[6736 → 7 :54]
8865	$\overline{\overline{\overline{\overline{3x}+x}+\overline{3x}+2x}+\overline{3x}}=\overline{\overline{3x}+x}+2x$	[8855 → 7]
8866	$\overline{\overline{\overline{3x}+x}+\overline{3x}}=x$	[8855 → 7: 11]
8870	$\overline{\overline{\overline{\overline{3x}+x}+\overline{3x}+y}+\overline{x+y}}=y$	[8866 → 7]
8871	$\overline{\overline{3x}+x}+2x=2x$	[8865 :8870]

Equation 8871 asserts the existence of an object c, namely $2x$, and an object d, namely $\overline{\overline{3x}+x}$, such that $c+d=c$. Q.E.D. (Note that 8855 satisfies $\exists c \exists d(\overline{c+d}=\overline{c})$.)

Step 6736 in Detail

Premises:

$$\overline{\overline{\overline{x+y}+\overline{x}+y}+y}=\overline{x+y} \qquad (10)$$

$$\overline{\overline{\overline{\overline{\overline{\overline{x}+y}+x+2y}+\overline{\overline{x}+y}+\overline{y+z}+z}+z+u}+\overline{\overline{y+z}+u}}=u \qquad (674)$$

With (10), let x be $3v$, and y be $\overline{\overline{3v}+v}+2v$:

$$\underbrace{\overline{\overline{\overline{3v+\overline{\overline{3v}+v}+2v}+\overline{3v}+\overline{\overline{3v}+v}+2v}+\overline{\overline{3v}+v}+2v}}=\overline{3v+\overline{\overline{3v}+v}+2v} \qquad (10')$$

With (674), let x be $3v$, y be v, z be $2v$, and u be $\overline{\overline{3v}+v}$:

$$\overline{\underbrace{\overline{\overline{\overline{\overline{\overline{3v}+v}+5v}+\overline{\overline{3v}+v}+\overline{3v}+2v}+2v+\overline{\overline{3v}+v}}}+\overline{\overline{3v}+\overline{\overline{3v}+v}}}=\overline{\overline{3v}+v} \qquad (674')$$

Replace the braced term of (674′), which is AC-identical to the left-hand side of (10′), with the right-hand side of (10′):

$$\overline{\overline{3v+\overline{\overline{3v}+v}+2v}+\overline{\overline{3v}+\overline{\overline{3v}+v}}}=\overline{\overline{3v}+v}.$$

Rename the variable and rearrange:

$$\overline{\overline{\overline{\overline{3x}+x}+\overline{3x}}+\overline{\overline{\overline{3x}+x}+5x}}=\overline{\overline{3x}+x}. \qquad [10 \rightarrow 674]\ (6736)$$

Single Axioms for Boolean Algebra

(McCune, Veroff, Fitelson, Harris, Feist, Wos, 2000)

- Problem: *Find a short single axiom for Boolean algebra in terms of disjunction and negation* $\{+,'\}$, i.e., join and complement.
- That is, find a short equation equivalent to

$x+y=y+x$	(C) Commutativity
$(x+y)+z=x+(y+z)$	(A) Associativity
$\overline{(\overline{(x+\overline{y})}+\overline{(x+y)})}=x$	(R) Robbins

- Previously known:
  - single axioms exist for this theory
  - there is a algorithm to build single axioms
  - the algorithm produces *very big* single axioms

A Method to Search for Single Axioms

We use Otter (theorem prover) and SEM (model searcher).

Maintain a set $\mathcal{S}$ of non-Boolean structures, e.g., $+$ is not commutative, or not idempotent.

Maintain a set $\mathcal{C}$ of candidate equations.

- enumerate well-formed equations
- for each equation E:
  - delete E if non-identity (decision procedure);
  - delete E if true in any member of $\mathcal{S}$;
  - search for a non-Boolean model M of E;
    if found,
    then insert M into $\mathcal{S}$,
    else insert E into $\mathcal{C}$;
- for each candidate in $\mathcal{C}$, use Otter to search for a proof of a known basis.

Searching for Single Axioms

- We used several variations of the preceding procedure.
  - parallel search on 15 computers
- About 2,000,000,000 equations were considered.
- Several weeks of computer time were used.
- Most of the search was for a 3-variable axiom.
- Success: *A single axiom with 4 variables:*

$$\overline{(\overline{(\overline{(x+y)}+z)}+\overline{\overline{(x+\overline{(\overline{z}+\overline{(z+u)})})}})}=z$$

- Open questions:
  - *Is there a shorter axiom?*
  - *Is there an axiom with 3 variables?*

Algebraic Geometry

An Inference Rule for Cubic Curves

- From a deep theorem in algebraic geometry, extract the equational essence for cubic curves.
- Implement the rule in the theorem prover Otter, and experiment.
- Investigate the power of the implementation.
- Investigate the power of the rule.
- New results.
- On this project,
  - Padmanabhan is the mathematician (expert on algebraic geometry),
  - McCune is the computer scientist (author of theorem prover Otter).

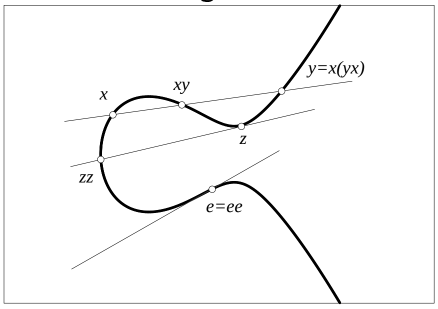
Basis for the Inference Rule

- A "basic" theorem in algebraic geometry:

  Let X be a projective curve and Y and Z be irreducible algebraic varieties, all defined over an algebraically closed field k. Let f be a regular mapping from $X \times Y$ into Z such that $f(X \times \{y_0\})$ is a singleton z_0 for some $y_0 \in Y$. Then $f(X \times \{y\})$ is a singleton for every $y \in Y$. [Shafarevich, p.152]

- Padmanabhan devised a first-order equational inference rule $=(gL)\Rightarrow$ that captures the above theorem for the algebra of cubic curves.
- $=(gL)\Rightarrow$ allows one to prove, in equational logic, some incidence theorems and configuration theorems on projective curves without reference to the underlying geometry or topology of the curves.
- $=(gL)\Rightarrow$ is unnatural for humans, but useful in programs.

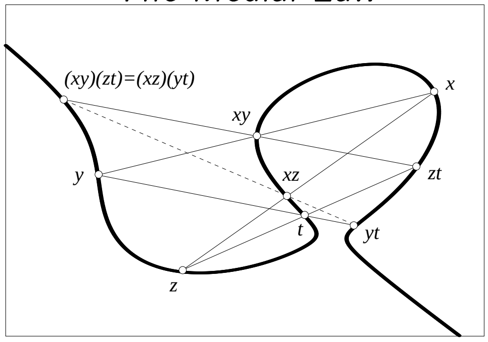
The Chord-Tangent Construction



Consider a non-singular cubic curve. Define a binary operation on the points of the curve.

- If a line intersects the curve in 3 points, the product of any two points is the third point.
- zz is the point where the tangent at z intersects the curve.
- Inflection points are idempotent ($ee = e$).
- The operation is commutative, and the Steiner law, $x(yx) = y$, holds.

The Medial Law



Theorem:

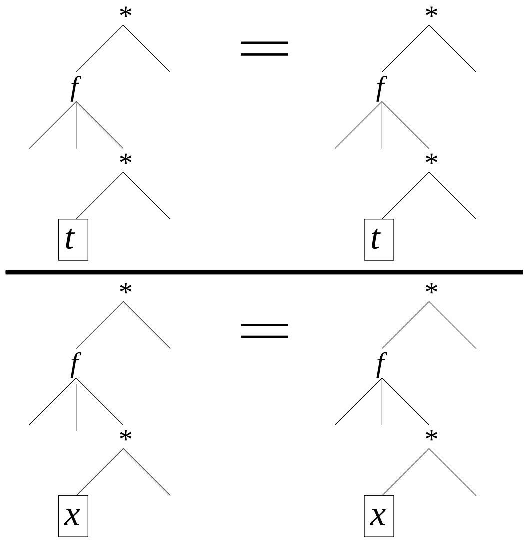
$$x(yx) = y \;=\!(gL)\!\Rightarrow\; (xy)(zu) = (xz)(yu).$$

Proof (Otter 3.0.3+ on gyro at 20.49 seconds).

2	$x \cdot (y \cdot x) = y$	
4	$(x \cdot (y \cdot x)) \cdot (z \cdot y) = z$	[2 → 2]
9	$(x \cdot y) \cdot x = y$	[2 → 2]
38	$(x \cdot (y \cdot x)) \cdot (z \cdot y) = (u \cdot z) \cdot u$	[9 → 4]
331	$(w \cdot (y \cdot x)) \cdot (z \cdot y) = (w \cdot z) \cdot x$	[gL 38]
333	$(w \cdot x) \cdot (z \cdot y) = (w \cdot z) \cdot (x \cdot y)$	[2 → 331]

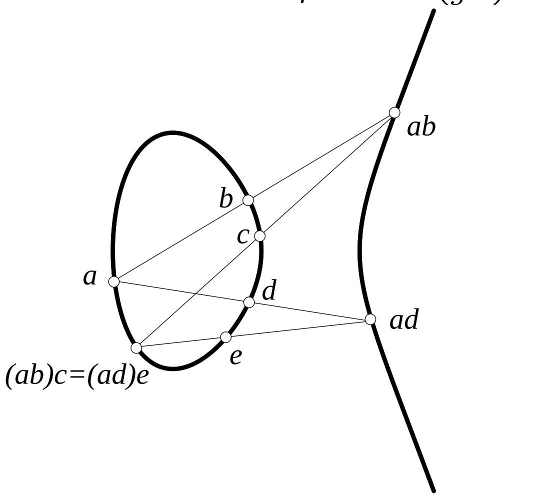
The proof ordinarily involves high-level concepts (the Bezout theorem, the Riemann-Roch theorem, or Weierstrass elliptic functions).

The Rule $=\!(gL)\!\Rightarrow$



Given an equality, if identical terms (t) occur in corresponding positions, and in the same nest of function symbols, replace both occurrences with a new variable (x).

A Geometric Example of $=\!(gL)\!\Rightarrow$



- $[(ab)c = (ad)e] \;=\!(gL)\!\Rightarrow\; [\forall x\ (xb)c = (xd)e]$.
- Select a point x and see if it works.

Soundness of $=\!(gL)\!\Rightarrow$

Assuming $=\!(gL)\!\Rightarrow$ is sound for the operation of the chord-tangent construction:

$$[(ab)c = (ad)e] \;=\!(gL)\!\Rightarrow\; [\forall x \ (xb)c = (xd)e]$$

- *Why is it sound for other operations?*
- We assume that other operations can be reduced to the chord-tangent operation.
- That is, all morphisms of the curve.
- For example, define a group operation on a cubic curve (example later).

Implementation of $=\!(gL)\!\Rightarrow$ *in Otter*

As an *inference rule*, with *unification*:

$$\frac{f(x,y,f(z,u,v,w),v) = f(x,y,v_6,f(z,u,v_6,w))}{f(x,y,v_0,v) = f(x,y,v_0,f(z,u,f(z,u,v,w),w))}$$

Details of inference:

– set $v_6 = f(z,u,v,w)$, by unification:

$$f(x,y,\underline{f(z,u,v,w)},v) = f(x,y,\underline{f(z,u,v,w)},f(z,u,f(z,u,v,w),w))$$

– replace underlined terms with new variable v_0.

Implementation of $=\!(gL)\!\Rightarrow$ *in Otter*

As a rewrite rule, applied to identical terms:

$$\frac{f(e,x,f(e,e,x)) = e}{f(v_1,x,f(v_2,v_3,x)) = f(v_1,y,f(v_2,v_3,y))}$$

Details of inference:

– Apply paramodulation:

$$f(e,x,f(e,e,x)) = f(e,y,f(e,e,y))$$

– Apply $=\!(gL)\!\Rightarrow$ 3 times, to the 3 positions of e, introducing 3 new variables:

$$f(\underline{v_1},x,f(\underline{v_2},\underline{v_3},x)) = f(\underline{v_1},y,f(\underline{v_2},\underline{v_3},y))$$

Strategies for $=\!(gL)\!\Rightarrow$ *(A)*

- *How does the rule* $=\!(gL)\!\Rightarrow$ *fit with our strategies for equational deduction?*
  - We must relax the rewrite-based strategy.
  - Note that terms in proof are not simplified.

$$\{ex = xe = x\} \;=\!(gL)\!\Rightarrow\; \{xy = yx\}.$$

Proof (Otter 3.0.3+ on gyro at 31.16 seconds).

2	$e \cdot x = x$	
3	$x \cdot e = x$	
5	$(e \cdot e) \cdot x = x$	[2 → 2]
7	$(x \cdot e) \cdot e = x$	[3 → 3]
79	$(x \cdot y) \cdot e = (e \cdot y) \cdot x$	[5 → 7 :gL]
355	$(e \cdot x) \cdot y = y \cdot x$	[3 → 79, flip]
814	$x \cdot y = y \cdot x$	[2 → 355]

Strategies for $=\!(gL)\!\Rightarrow$ *(B)*

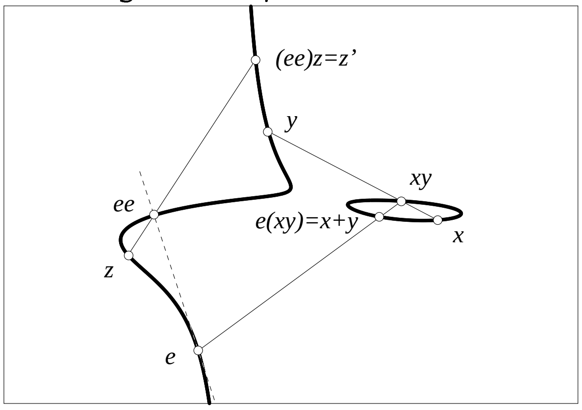
In some cases, we can rewrite everything.

$$\left\{\begin{array}{l} x(yx) = y \\ ex = xe \\ ee = e \end{array}\right\} =\!(gL)\!\Rightarrow \{(x((z(xy))(ey)))(ee) = z\}$$

Proof (Otter 3.0.3+ on gyro at 10.12 seconds).

1	$x = x$	
3,2	$e \cdot e = e$	
4	$e \cdot x = x \cdot e$	
6,5	$x \cdot (y \cdot x) = y$	
7	$(A \cdot ((C \cdot (A \cdot B)) \cdot (e \cdot B))) \cdot (e \cdot e) \neq C$	
8	$(A \cdot ((C \cdot (A \cdot B)) \cdot (e \cdot B))) \cdot e \neq C$	[copy,7 :3]
18,17	$(x \cdot y) \cdot x = y$	[5 → 5]
20,19	$e \cdot (e \cdot x) = x$	[4 → 5]
21	$x \cdot (x \cdot e) = e$	[4 → 5]
34	$x \cdot (x \cdot y) = y$	[19 → 21 :gL, 20]
41,40	$(x \cdot y) \cdot y = x$	[17 → 34]
42	$x \cdot y = y \cdot x$	[5 → 34]
46	$(x \cdot y) \cdot z = z \cdot (y \cdot x)$	[42 → 42]
53	$(A \cdot ((C \cdot (B \cdot A)) \cdot (e \cdot B))) \cdot e \neq C$	[42 → 8]
114	$(x \cdot y) \cdot (z \cdot u) = (u \cdot z) \cdot (y \cdot x)$	[42 → 46]
204	$(x \cdot y) \cdot (z \cdot u) = (x \cdot z) \cdot (y \cdot u)$	[gL 114]
231	$C \neq C$	[204 → 53 :18,6,41]
232	$\square$	[231,1]

Defining a Group on a Cubic Curve



- '$\cdot$' is the operation of chord-tangent construction.
- Select any point as the identity e.
- Define $x + y$ as $(x \cdot y) \cdot e$.
- Define $-x$ as $(e \cdot e) \cdot x$.

With a given identity, the group is unique.

Uniqueness of Inversive Groupoids (A)

$$\left\{\begin{array}{ll} f(e,x) = x, & F(e,x) = x \\ f(x,e) = x, & F(x,e) = x \\ f(g(x),x) = e, & F(G(x),x) = e \\ f(x,g(x)) = e, & F(x,G(x)) = e \end{array}\right\} =\!(gL)\!\Rightarrow$$

$$\left\{\begin{array}{l} f(x,y) = F(x,y) \\ g(x) = G(x) \end{array}\right\}.$$

- A rewrite strategy that works for this problem.
  - Do not use input equations as rewrite rules.
  - Use instances of input equations and some lemmas as rewrite rules:

    $f(e,e) = e, \quad F(e,e) = e, \quad g(e) = e, \quad G(e) = e$
  - Make orientable derived equations into rewrite rules.
  - Paramodulate from both sides into both sides.
  - Paramodulate from variables but not into variables.

Uniqueness of Inversive Groupoids (B)

Proof (Otter 3.0.3e on gyro at 11.77 seconds).

2	$f(e,x) = x$	
3	$f(x,e) = x$	
4	$f(g(x),x) = e$	
6	$F(e,x) = x$	
7	$F(x,e) = x$	
9	$F(x,G(x)) = e$	
10	$f(A,B) = F(A,B),\ g(A) = G(A) \ \rightarrow\ \square$	
18,17	$f(f(x,e),e) = x$	[3 → 3]
21	$f(f(e,x),e) = x$	[3 → 2]
26,25	$f(F(e,x),e) = x$	[3 → 6]
38,37	$f(F(x,e),e) = x$	[3 → 7]
47	$f(F(g(x),e),x) = e$	[7 → 4]
49	$f(F(e,g(x)),x) = e$	[6 → 4]
53	$f(f(e,g(x)),x) = e$	[2 → 4]
63	$f(g(x),x) = f(g(y),y)$	[4 → 4]
71,70	$f(f(g(x),x),y) = y$	[4 → 2]
154,153	$f(F(x,G(x)),y) = y$	[9 → 2]
163	$f(F(g(x),y),x) = y$	[37 → 47 :gL, 26]
173	$G(g(x)) = f(e,x)$	[9 → 163, flip]
199	$f(F(x,g(y)),y) = x$	[37 → 49 :gL, 38]
294	$f(f(x,g(y)),y) = x$	[21 → 53 :gL, 18]
307	$F(x,g(g(y))) = f(x,y)$	[199 → 294, flip]
335,334	$g(g(x)) = x$	[63 → 294 :71, flip]
337	$f(x,y) = F(x,y)$	[307 :335, flip]
604	$G(g(x)) = x$	[9 → 173 :154]
608	$g(x) = G(x)$	[334 → 604, flip]
627	$\square$	[608,10,337]

Cancellative Semigroups

Cancellative $=\!(gL)\!\Rightarrow$ -semigroups are commutative.

$$\left\{ \begin{array}{l} \text{cancellation} \\ (xy)z = x(yz) \end{array} \right\} =\!(gL)\!\Rightarrow\ xy = yx$$

Proof (Otter 3.0.4 on gyro at 6.13 seconds):

2	$xy = u,\ xz = u\ \rightarrow\ y = z$	
3	$yx = u,\ zx = u\ \rightarrow\ y = z$	
4	$(xy)z = x(yz)$	
5	$BA = AB\ \rightarrow\ \square$	
6	$(((xy)z)u)v = (x(yz))(uv)$	[4 → 4]
11	$(x(yz))u = (xy)(zu)$	[4 → 4]
432	$(((xy)z)u)v = (xu)((yz)v)$	[(gL) 6]
433	$((xy)z)u = x(u(yz))$	[3,11,432, flip]
1205	$xy = yx$	[2,432,433]
1206	$\square$	[1205,5]

n-ary Steiner Laws

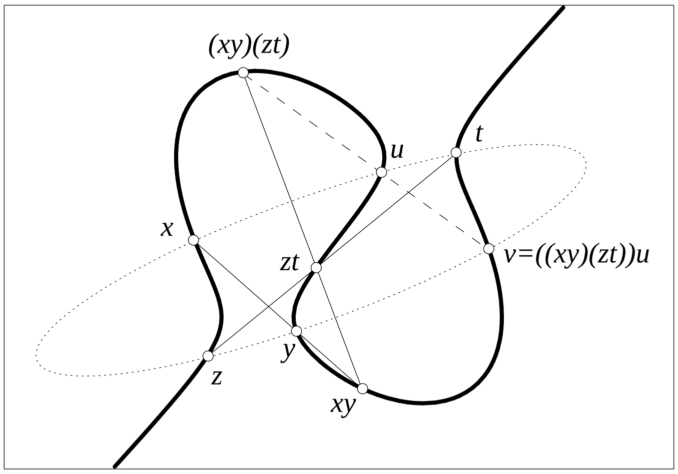
- Uniqueness of the 5-ary Steiner Law:

$$\left\{ \begin{array}{l} \text{full permutability of } f \\ f(e,e,e,e,e) = e \\ f(e,e,e,x,f(e,e,e,x,y)) = y \\ \text{full permutability of } g \\ g(e,e,e,e,e) = e \\ g(e,e,e,x,g(e,e,e,x,y)) = y \end{array} \right\} =\!(gL)\!\Rightarrow$$

$$g(x,y,z,u,v) = f(x,y,z,u,v).$$

A corollary: given 5 points of intersection of a cubic and a conic, the sixth point can be found be a simple ruler construction.

Conic-Cubic Construction



Summary

Programs such as Otter, EQP, MACE, and SEM serve as symbolic calculators and as deductive assistants.

- They apply to problems of interest to mathematicians.
- They work in ways that are unnatural to humans.
  - more search, less insight
- They find unnatural solutions and proofs.
- In some cases, they find

  new and interesting results.

References: `www.mcs.anl.gov/~mccune`