Theorem Provers and Computer Algebra Systems

John Harrison

Cambridge University Computer Laboratory

2nd November 1994

Theorem Provers

- **Are mainly used by computer scientists**

- **Applications include hardware, software and protocol verification**

- **Aim to support logic as applied mathematics**

- **Generally use "discrete" mathematics**

Computer Algebra Systems

- **Are mainly used by applied mathematicians, engineers and scientists**

- **Multiprecision arithmetic, differentiation, integration . . .**

- **Aim to support conventional applied mathematics**

- **Mainly use "continuous" mathematics**

Features of Theorem Provers

- **They are logically and mathematically precise**

- **They employ rigorous principles of deduction**

- **They are usually difficult to use**

- **They are often very slow**

Computer Algebra Systems

- **Are easy to use**
- **Are efficient and powerful**
- **Lack a precise notion of logic**
- **Are deductively unsound**

The Lack of Logic in Computer Algebra Systems

They are mainly based on a simple dialogue with the user:

- **The user gives an expression** E_1
- **The CAS returns an expression** E_2
- **We are supposed to believe that** $E_1 = E_2$

But are we? What about undefinedness?

$$\frac{x^2 - 1}{x - 1} = x + 1$$

Sometimes we can reason about simple inequalities, and there is at least a case analysis . . .

The Unsoundness of Computer Algebra Systems

- **Maple:**

$$\int_{-1}^{1} \sqrt{x^2}\ dx = 0$$

- **Mathematica:**

$$\int_{-1}^{1} \frac{1}{\sqrt{x^2}}\ dx = 0$$

Anyway is an antiderivative what we want? Maybe we want

- **Riemann Integral**
- **Lebesgue Integral**
- **Gauge Integral**

The Spectrum of Theorem Proving Systems

- **Proof Checkers**
  - **Automath (de Bruijn)**
  - **Stanford LCF (Milner et al.)**

. . .

. . .

. . .

- **Automatic Theorem Provers**
  - **NQTHM (Boyer-Moore)**
  - **Otter (McCune)**

Which approach is better?

The LCF approach

Aims to combine low-level proof checker and high level theorem prover.

- Low-level primitive inferences

- Use of ML as programming environment for writing complex procedures

- Secure abstract datatype of theorems

The LCF family

- **Original was Edinburgh LCF (Milner, Gordon, Morris, Newey, Wadsworth)**

- **Reengineered as Cambridge LCF (Paulson)**

- **Many descendants include**
  - **HOL (Gordon)**
  - **Nuprl (Constable)**
  - **Coq (Huet)**

- **Refinements of the basic idea include Isabelle (Paulson)**

The ML programming language started life as the MetaLanguage for LCF

Quick Summary of HOL

- **Higher order logic based on simply typed lambda calculus**

- **ML-style parametric polymorphism**

- **Conservative definition mechanism**

- **Very few primitive rules (in theory)**

- **Several versions (HOL88, hol90, ProofPower)**

Analytica – a remedy for the lack of logic

- Designed by Clarke and Zhao

- Written in the Mathematica language

- Incorporates many powerful decision procedures

- But it relies on Mathematica's own (unsound) simplifier

Mathpert – a remedy for the lack of soundness

- Designed by Beeson

- Intended for educational use; stresses 'glass box' approach

- Underlying sequent calculus where side conditions accumulate

- Attempt to avoid the logic appearing explicitly

- It remains to be seen how it compares with existing systems in power

Harrison and Théry – exploiting a link

We link together a Theorem Prover (HOL) and a Computer Algebra System (Maple).

HOL can ask Maple questions – but what do we do with the answers?

1. **Trust the Computer Algebra System completely**

2. **Trust it partially; tag the theorem**

3. **Don't trust it at all – check the answer**

Examples where Checking is Easy

- **Solving equations (of all kinds)**

- **Factorizing polynomials (or indeed numbers!)**

- **Integrating expressions**

Example combining integration and factorization (1)

We want to evaluate:

$$\int_0^t \sin^3 u \; du$$

Maple tells us:

$$\int_0^t \sin^3 u \; du = -\frac{1}{3}\sin^2 t \cos t - \frac{2}{3}\cos t + \frac{2}{3}$$

HOL can differentiate this expression to yield

$$-\frac{1}{3}\left(2 \sin t \cos t \cos t - \sin^3 t\right) + \frac{2}{3}\sin t$$

but it doesn't simplify down to what we wanted (neither does Maple in fact!)

Example combining integration and factorization (2)

We want to show that

$$-\frac{1}{3}\left(2\,\mathbf{sin}\,t\,\mathbf{cos}\,t\,\mathbf{cos}\,t - \mathbf{sin}^3 t\right) + \frac{2}{3}\,\mathbf{sin}\,t = \mathbf{sin}^3\,t$$

Let's replace $\mathbf{sin}\,t$ by x and $\mathbf{cos}\,t$ by y; we want to show that

$$\vdash -\frac{1}{3}\left(2\,x\,y\,y - x^3\right) + \frac{2}{3}\,x - x^3 = 0$$

Example combining integration and factorization (3)

We ask Maple to factorize this expression, and it tells us:

$$\vdash -\frac{1}{3}(2\,x\,y\,y - x^3) + \frac{2}{3}\,x - x^3 = -\frac{2}{3}\,x\,(y^2 + x^2 - 1)$$

HOL can check this answer very easily.

When $x = \sin t$ **and** $y = \cos t$ **we have** $y^2 + x^2 - 1 = 0$**, so the equation is proved.**

Now the Fundamental Theorem of Calculus yields the result. Maple was right!

What have we Gained?

In HOL, real analysis, including (gauge) integration and its relationship with differentiation, has been developed formally by definitional means. So we have:

- An independent check on Maple's correctness

- A formal HOL proof using incontrovertible, low-level principles

- A rigorously defined, mathematically useful statement

Conclusions

- **More experience needed. Does rigour mean rigor mortis?**

- **For the approach to generalize, we need powerful simplifiers**

- **But it gives quite a lot for very little work**

- **Theorem prover and computer algebra designers have a lot to learn from each other.**