two automation challenges

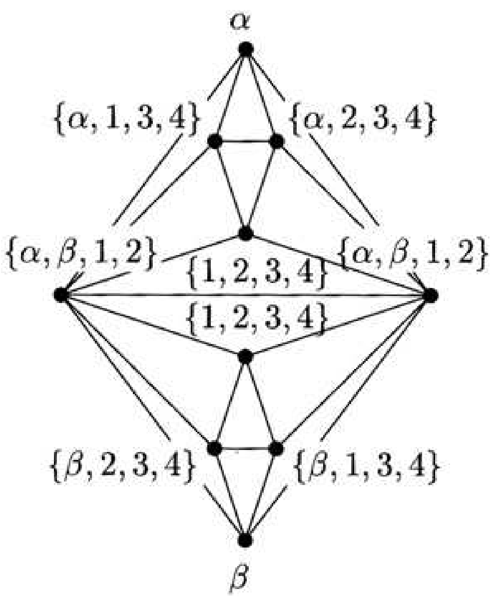
Freek Wiedijk

Radboud University Nijmegen

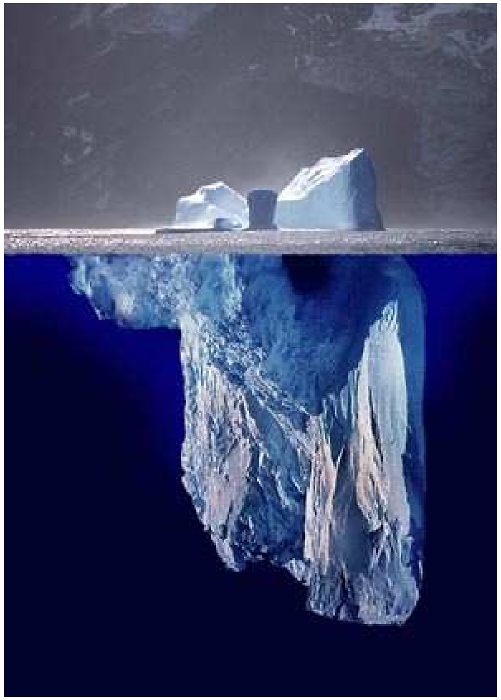
Dagstuhl seminar: interaction versus automation

2009 10 06, 17 : 00

program verification versus mathematics



assistant versus word processor



textbook proof from Hardy & Wright

Theorem 43 (Pythagoras' theorem). $\sqrt{2}$ is irrational.

The traditional proof ascribed to Pythagoras runs as follows. If $\sqrt{2}$ is rational, then the equation

$$a^2 = 2b^2 \tag{4.3.1}$$

is soluble in integers a, b with $(a, b) = 1$. Hence a^2 is even, and therefore a is even. If $a = 2c$, then $4c^2 = 2b^2$, $2c^2 = b^2$, and b is also even, contrary to the hypothesis that $(a, b) = 1$. $\square$

formal proof sketch of the textbook proof

theorem Th43: sqrt 2 is irrational :: **Pythagoras' theorem**

proof assume sqrt 2 is rational; consider a, b such that

4_3_1: $$a\hat{\ }2 = 2 * b\hat{\ }2$$

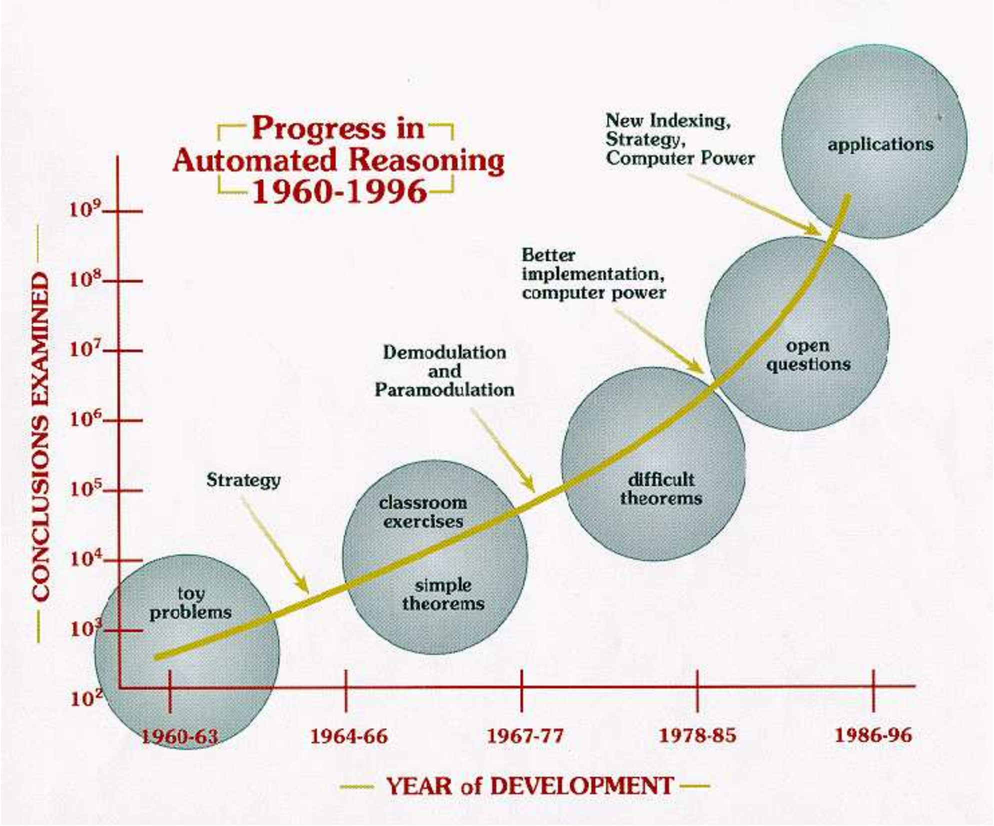
and a, b are_relative_prime; $a\hat{\ }2$ is even; a is even; consider c such that $a = 2 * c$; $4 * c\hat{\ }2 = 2 * b\hat{\ }2$; $2 * c\hat{\ }2 = b\hat{\ }2$; b is even; thus contradiction; end;

full declarative formalization

```
theorem Th43: sqrt 2 is irrational
proof
  assume sqrt 2 is rational;
  then consider a, b such that
A1: b <> 0 and
A2: sqrt 2 = a/b and
A3: a, b are_relative_prime by Def1;
A4: b^2 <> 0 by A1, SQUARE_1:73;
  2 = (a/b)^2 by A2, SQUARE_1:def 4
    .= a^2/b^2 by SQUARE_1:69;
  then
4_3_1: a^2 = 2 * b^2 by A4, REAL_1:43;
  a^2 is even by 4_3_1, ABIAN:def 1;
  then
A5: a is even by PYTHTRIP:2;
:: continue in next column
  then consider c such that
A6: a = 2 * c by ABIAN:def 1;
A7: 4 * c^2 = (2 * 2) * c^2
      .= 2^2 * c^2 by SQUARE_1:def 3
      .= 2 * b^2 by A6, 4_3_1, SQUARE_1:68;
  2 * (2 * c^2) = (2 * 2) * c^2 by AXIOMS:16
      .= 2 * b^2 by A7;
  then 2 * c^2 = b^2 by REAL_1:9;
  then b^2 is even by ABIAN:def 1;
  then b is even by PYTHTRIP:2;
  then 2 divides a & 2 divides b by A5, Def2;
  then
A8: 2 divides a gcd b by INT_2:33;
  a gcd b = 1 by A3, INT_2:def 4;
  hence contradiction by A8, INT_2:17;
end;
```

first challenge: automate elementary reasoning steps

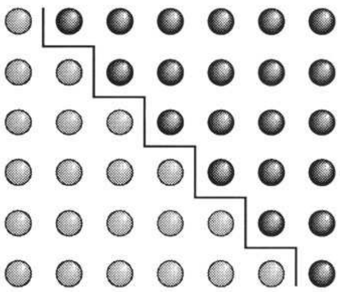
Larry Wos:



- experience with formal proof sketches:
  computers routinely proving non-trivial steps is far away
- focus should be on making manual math formalization efficient

luxury mathmode

procedural proof using tactics



```
# g `!n. nsum(1..n) (\i. i) = (n*(n + 1)) DIV 2`;;
val it : goalstack = 1 subgoal (1 total)

`!n. nsum (1..n) (\i. i) = (n * (n + 1)) DIV 2`

# e INDUCT_TAC;;
val it : goalstack = 2 subgoals (2 total)

  0 [`nsum (1..n) (\i. i) = (n * (n + 1)) DIV 2`]

`nsum (1..SUC n) (\i. i) = (SUC n * (SUC n + 1)) DIV 2`

`nsum (1..0) (\i. i) = (0 * (0 + 1)) DIV 2`

# e (ASM_REWRITE_TAC[NSUM_CLAUSES_NUMSEG]);;
val it : goalstack = 1 subgoal (2 total)

`(if 1 = 0 then 0 else 0) = (0 * (0 + 1)) DIV 2`

#
```

batch checked declarative proof

```
!n. nsum(1..n) (\i. i) = (n*(n + 1)) DIV 2
proof
  nsum(1..0) (\i. i) = 0 by NSUM_CLAUSES_NUMSEG;
    ... = (0*(0 + 1)) DIV 2 [1];
  now let n be num;
    assume nsum(1..n) (\i. i) = (n*(n + 1)) DIV 2 [2];
    1 <= SUC n;
    nsum(1..SUC n) (\i. i) = (n*(n + 1)) DIV 2 + SUC n
      by NSUM_CLAUSES_NUMSEG,2;
    thus ... = ((SUC n)*(SUC n + 1)) DIV 2;
  end;
qed by INDUCT_TAC,1;
```

Mizar Light

= ‘luxury mathmode’ (Henk)

= proof language/interface on top of HOL Light

demo

computer algebra with assumptions

two flavors of computer algebra

mathematical computation by computer

numerical computation

symbolic computation

computer algebra

computer calculus

$$\frac{1}{X} \in \mathbb{C}(X)$$

$$\lambda X.\, \frac{1}{X} \in \mathbb{C}^{\mathbb{C}_{\neq 0}}$$

$\frac{X}{X} = 1$ as algebraic objects

$\frac{X}{X} \neq 1$ when $X = 0$

the Content MathML signature

147 XML elements, like:

$$ {}^{-1} \quad \lambda \quad \circ \quad ! \quad \div \quad \max \quad \min \quad - \quad + \quad \cdot \quad \sqrt{\ } \quad \gcd \quad \wedge \quad \vee \quad \neg \quad \Rightarrow \quad \forall \quad \exists \quad |\cdot| \quad \overline{\cdot} \quad \arg $$

$$ \Re \quad \Im \quad \lfloor\cdot\rfloor \quad \lceil\cdot\rceil \quad = \quad \neq \quad > \quad < \quad \geq \quad \leq \quad \Leftrightarrow \quad \approx \quad | \quad \int \quad \frac{d}{dx} \quad \frac{\partial}{\partial x} \quad \nabla \quad \cup \quad \cap \quad \in \quad \subseteq $$

$$ \subset \quad \setminus \quad \# \quad \times \quad \sum \quad \prod \quad \lim \quad \ln \quad \log \quad \sin \quad \cos \quad \tan \quad \sec \quad \csc \quad \cot \quad \sinh \quad \cosh $$

$$ \tanh \quad \coth \quad \arcsin \quad \arccos \quad \arctan \quad \mu \quad \sigma \quad \det \quad {}^{\mathrm{T}} \quad \otimes \quad \mathbb{Z} \quad \mathbb{R} \quad \mathbb{Q} \quad \mathbb{N} \quad \mathbb{C} \quad e \quad i $$

$$ \top \quad \bot \quad \emptyset \quad \pi \quad \gamma \quad \infty $$

	MathML	LaTeX	HOL
$p \Rightarrow q$	`implies`	`\Rightarrow`	`==>`
$A \times B$	`cartesianproduct`	`\times`	`prod` , `CROSS`
0	`cn`		`NUMERAL`
$\sum_{i=1}^{n} a_i$	`sum`	`\sum`	`nsum` , `sum`
∞	`infinity`	`\infty`	

sample problem

$$x \neq 0 \;\wedge\; \bigl|\ln(x^2)\bigr| > 1 \;\wedge\; \int_0^x t\,dt \leq 1 \;\Rightarrow\; -\frac{1}{\sqrt{e}} < x < \frac{1}{\sqrt{e}}$$

- *this is not about first order proof search*

  first order proof search cannot easily calculate integrals
  first order proof search cannot easily do numerical approximations

- *this is not about decision procedures*

  decision procedures generally work over specific small signatures

- *this is not about systems like Maple and Mathematica*

  current computer algebra systems generally do not use assumptions

second challenge: take next step in mathematics automation

progress:

- automation of formalized primary school math = 'arithmetic'
- automation of formalized high school math = 'calculus'
- automation of formalized university math

should run in less than a second

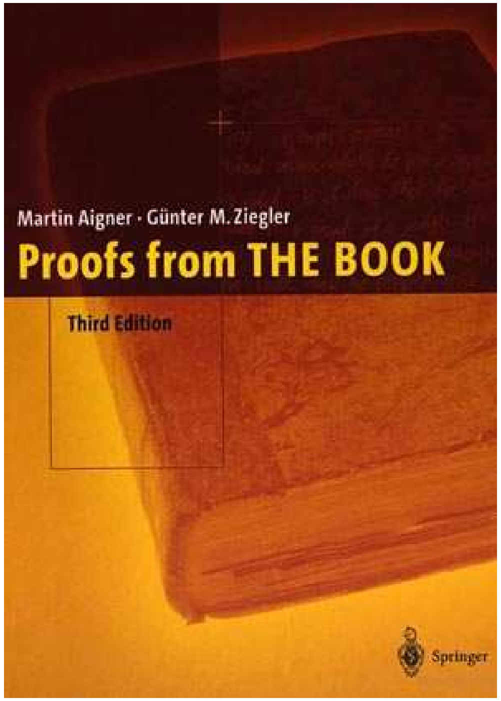
should run without any arguments
should implicitly use:

- assumptions in the goal = local labels in the proof
- theorems from the formal library

the plan

creating a collection of mathematical problems

- take proofs from **Proofs from The Book**
- create **formal proof sketches** of those proofs
- calculate the **proof obligations** of the steps in those formal proof sketches
- select proof obligations in the **Content MathML** signature



benchmark for math automation

the proof obligations for the Hardy & Wright example

$$
\begin{aligned}
\sqrt{2} \in \mathbb{Q} &\quad\vdash\quad \exists\, a, b \in \mathbb{Z}\, (a^2 = 2b^2 \;\wedge\; \mathsf{gcd}(a, b) = 1) \\
b \in \mathbb{Z} \;\wedge\; a^2 = 2b^2 &\quad\vdash\quad 2 \mid a^2 \\
a \in \mathbb{Z} \;\wedge\; 2 \mid a^2 &\quad\vdash\quad 2 \mid a \\
2 \mid a &\quad\vdash\quad \exists\, c \in \mathbb{Z}\, (a = 2c) \\
a^2 = 2b^2 \;\wedge\; a = 2c &\quad\vdash\quad 4c^2 = 2b^2 \\
4c^2 = 2b^2 &\quad\vdash\quad 2c^2 = b^2 \\
b \in \mathbb{Z} \;\wedge\; c \in \mathbb{Z} \;\wedge\; 2c^2 = b^2 &\quad\vdash\quad 2 \mid b \\
\mathsf{gcd}(a, b) = 1 \;\wedge\; 2 \mid a \;\wedge\; 2 \mid b &\quad\vdash\quad \bot
\end{aligned}
$$

Content MathML signature

only relevant subset of the assumptions shown here

each should be proved automatically in less than a second!