

Proving with Computer Assistance

Lecture 1

Herman Geuvers

Administration

- ▶ Teacher: Herman Geuvers (Thursday only, HG 6.88)
- ▶ Mail to `herman@cs.ru.nl`
- ▶ Web page:
`http://www.cs.ru.nl/H.Geuvers/onderwijs/provingwithCA/`
- ▶ Weekly overview: see the webpage
- ▶ For practical work in the Proof Assistant Coq, we want to create logins for you, so you can work with Coq through a web-interface.

Content

- ▶ Logic, Natural Deduction (known?)
- ▶ Lambda calculus (known?)
- ▶ Type Theory
- ▶ Working with the Proof Assistant Coq
- ▶ Working with the Proof Assistant PVS

The general picture

What are Proof Assistants for?

The general picture

What are Proof Assistants for?

- ▶ Precise mathematical modelling (defining)
- ▶ Verification of properties of systems (proving)

The general picture

What are Proof Assistants for?

- ▶ Precise mathematical modelling (defining)
- ▶ Verification of properties of systems (proving)

Computer supports in these activities:

- ▶ Checking correctness of definitions
- ▶ Take care of the bookkeeping
- ▶ Do some computation
- ▶ Do some proving for us

The general picture

Does the Proof Assistant do all the proving for us?

The general picture

Does the Proof Assistant do all the proving for us?

No ...

It is undecidable in general whether a certain formula is true or not.

Automated Theorem Provers	Proof Assistants
Specific domains	Generally applicable
Massage your problem	Modelling is direct
False or True (with a proof?)	Interactive, user guided

The general picture

- ▶ Automated Theorem Provers
E.g. Otter, ACL2 ... Specialized (e.g. logic programs), Built-in automation (e.g. resolution)
- ▶ Model Checkers
E.g. Uppaal, Spin, SMV ... Specialized (reachability problems), Built-in automation (state space abstraction)
- ▶ Computer Algebra Systems
E.g. Maple, Mathematica ... Specialized (solving equations over \mathbf{C}), Built-in automation (symbolic term rewriting), may give wrong answer.
- ▶ Proof Assistants
E.g. Coq, PVS, Mizar, Hol (light), Isabelle ... Generic, Little automation (program your own ...)

Use of PAs

Who is using Proof Assistants and what for?

Use of PAs

Who is using Proof Assistants and what for?

Computer Scientists for

- ▶ Modelling and specifying systems
- ▶ Proving the correctness of models / software / systems

Use of PAs

Who is using Proof Assistants and what for?

Computer Scientists for

- ▶ Modelling and specifying systems
- ▶ Proving the correctness of models / software / systems

Mathematicians ??

Use of PAs

Who is using Proof Assistants and what for?

Computer Scientists for

- ▶ Modelling and specifying systems
- ▶ Proving the correctness of models / software / systems

Mathematicians ?? for

- ▶ Building up theories
- ▶ Verifying proofs

Use of PAs

Who is using Proof Assistants and what for?

Computer Scientists for

- ▶ Modelling and specifying systems
- ▶ Proving the correctness of models / software / systems

Mathematicians ?? for

- ▶ Building up theories
- ▶ Verifying proofs

Mathematicians are not big users of Proof Assistants

- ▶ Mechanically verifying a proof takes too much time. (Too much idiosyncrasy, not enough automation.)

Use of PAs

Who is using Proof Assistants and what for?

Computer Scientists for

- ▶ Modelling and specifying systems
- ▶ Proving the correctness of models / software / systems

Mathematicians ?? for

- ▶ Building up theories
- ▶ Verifying proofs

Mathematicians are not big users of Proof Assistants

- ▶ Mechanically verifying a proof takes too much time. (Too much idiosyncrasy, not enough automation.)
- ▶ We don't need computers to verify proofs! We are much better at it!

Mathematical users of Proof Assistants

Mathematicians are skeptical, but there are exceptions . . . and young mathematicians are less afraid of computers . . .

Mathematical users of Proof Assistants

Mathematicians are skeptical, but there are exceptions ... and young mathematicians are less afraid of computers ...

- ▶ Store formalized mathematics on a computer ... make it *actively* available for proof assistants, automated theorem provers, computer algebra systems, presentation tools, editors, ...
- ▶ Mathematical Knowledge Management (MKM): an emerging research field.

Mathematical users of Proof Assistants

Mathematicians are skeptical, but there are exceptions ... and young mathematicians are less afraid of computers ...

- ▶ Store formalized mathematics on a computer ... make it *actively* available for proof assistants, automated theorem provers, computer algebra systems, presentation tools, editors, ...
- ▶ Mathematical Knowledge Management (MKM): an emerging research field.
- ▶ Use Web technology, XML, MathML, OpenMath, ...

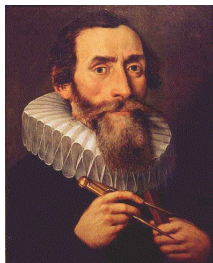
Mathematical users of Proof Assistants

Flyspeck project: Formalizing a proof of the Kepler Conjecture

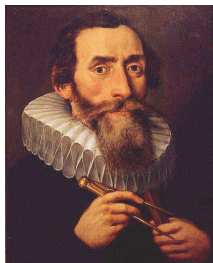
<http://code.google.com/p/flyspeck/>

Tom Hales, CMU Pittsburgh

Kepler Conjecture (1611)

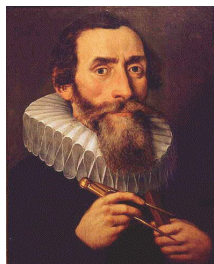


Kepler Conjecture (1611)

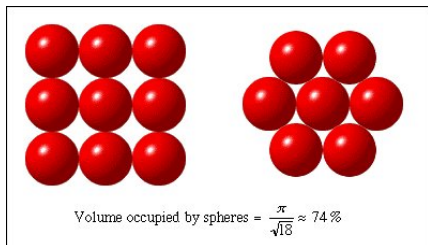


The most compact way of stacking balls of the same size is a pyramid.

Kepler Conjecture (1611)



The most compact way of stacking balls of the same size is a pyramid.



Kepler Conjecture (1611)

- ▶ Hales 1998: proof of the conjecture using computer programs (300 pages)



Thomas Hales, associate professor of mathematics, demonstrates "his solution" to the Kepler conjecture, a problem that mathematicians have been wrestling with since 1611. (Image courtesy of the Virginia Tennis Club. Photo by Bob Kambsco)"/>Thomas Hales, associate professor of mathematics, demonstrates "his solution" to the Kepler conjecture, a problem that mathematicians have been wrestling with since 1611. (Image courtesy of the Virginia Tennis Club. Photo by Bob Kambsco)

- ▶ Annals of Mathematics: 99% correct ...

Kepler Conjecture (1611)

- ▶ Hales 1998: proof of the conjecture using computer programs (300 pages)



Thomas Hales, associate professor of mathematics, demonstrates "his solution" to the Kepler conjecture, a problem that mathematicians have been wrestling with since 1611. (Image courtesy of the Virginia Tennis Club. Photo by Bob Kambsco) "

- ▶ Annals of Mathematics: 99% correct . . . but we can't verify the correctness of the computer programs.

Hales' proof of the Kepler conjecture

Reduce the problem to 1039 inequalities of the shape

$$\frac{-x_1x_3 - x_2x_4 + x_1x_5 + x_3x_6 - x_5x_6 + x_2(-x_2 + x_1 + x_3 - x_4 + x_5 + x_6)}{\sqrt{4x_2 \left(\begin{array}{l} x_2x_4(-x_2 + x_1 + x_3 - x_4 + x_5 + x_6) + \\ x_1x_5(x_2 - x_1 + x_3 + x_4 - x_5 + x_6) + \\ x_3x_6(x_2 + x_1 - x_3 + x_4 + x_5 - x_6) \\ -x_1x_3x_4 - x_2x_3x_5 - x_2x_1x_6 - x_4x_5x_6 \end{array} \right)}} < \tan\left(\frac{\pi}{2} - 0.74\right)$$

Hales' proof of the Kepler conjecture

Reduce the problem to 1039 inequalities of the shape

$$\frac{-x_1x_3 - x_2x_4 + x_1x_5 + x_3x_6 - x_5x_6 + x_2(-x_2 + x_1 + x_3 - x_4 + x_5 + x_6)}{\sqrt{4x_2 \left(\begin{array}{l} x_2x_4(-x_2 + x_1 + x_3 - x_4 + x_5 + x_6) + \\ x_1x_5(x_2 - x_1 + x_3 + x_4 - x_5 + x_6) + \\ x_3x_6(x_2 + x_1 - x_3 + x_4 + x_5 - x_6) \\ -x_1x_3x_4 - x_2x_3x_5 - x_2x_1x_6 - x_4x_5x_6 \end{array} \right)}} < \tan\left(\frac{\pi}{2} - 0.74\right)$$

Use computer programs to verify these inequalities.

Flyspeck project

- ▶ Hales: **formalise** the proof of Kepler's conjecture using **Proof Assistants** Write the computer code in the PA, prove it correct in the PA and run it in the PA.
- ▶ Proof Assistants used: Hol light, Isabelle, Coq

Computer Science users of Proof Assistants

Verify software and systems

- ▶ Popl mark challenge: Mechanized metatheory for the masses.
Formally prove the meta theory of your programming language
- ▶ Compiler correctness in Coq
Cristal project (X. Leroy): proving a C-compiler correct in Coq
- ▶ Java Bytecode verification (Isabelle, Coq)

Some history of Proof Assistants

- ▶ Church 1940 λ -calculus, simple type theory, higher order logic
- ▶ Curry Howard (De Bruijn): Formulas-as-Types
Encode proofs as terms, represent formulas as types
Proof-checking = Type-checking
- ▶ Automath (De Bruijn): first implementation of these ideas
- ▶ LCF (Milner), ML
- ▶ Coq, Hol, Isabelle, Mizar, PVS

These lectures

- ▶ Lambda calculus
See the notes by Rob Nederpelt. See the website for additional notes (including a short “crash course”)
- ▶ Type Theory
Is not only used for Proof Assistants but also very much in Programming Languages. In the lectures I'll devote attention to this. (Type checking algorithm, ...)
- ▶ Working with the Proof Assistant Coq
- ▶ Working with the Proof Assistant PVS