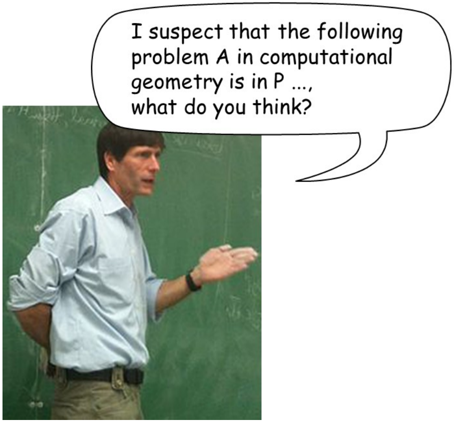
AI & Reasoning: A View From the Trenches

Josef Urban

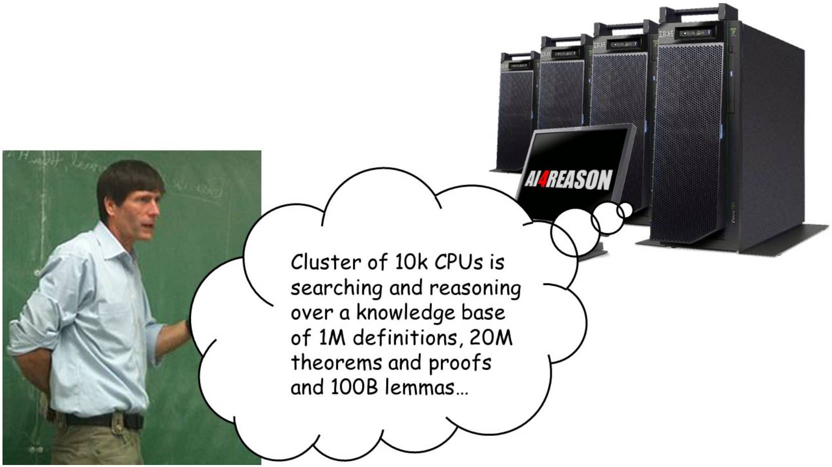
Czech Institute for Informatics, Robotics and Cybernetics
Czech Technical University in Prague

European Research Council
Established by the European Commission

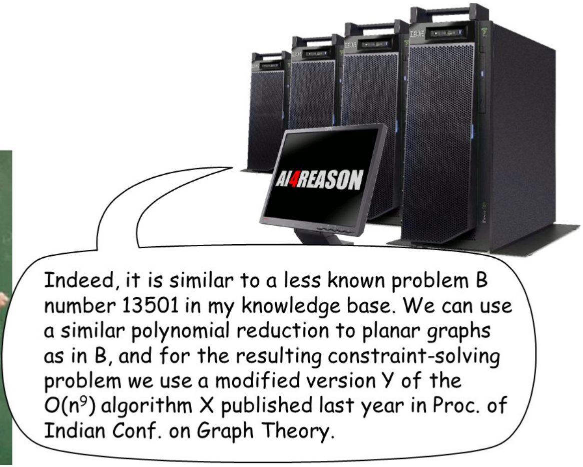
Not So Distant Future


I suspect that the following problem A in computational geometry is in P ..., what do you think?

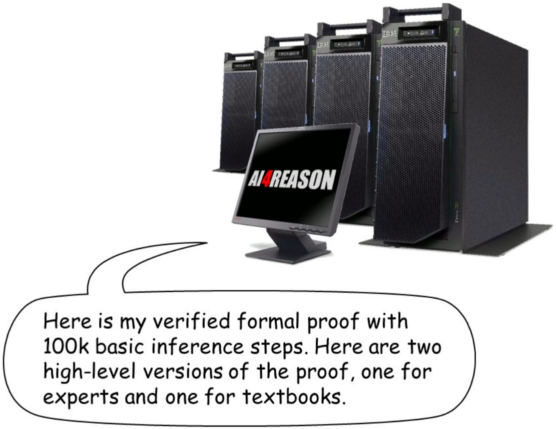
Not So Distant Future


AI4REASON
Cluster of 10k CPUs is searching and reasoning over a knowledge base of 1M definitions, 20M theorems and proofs and 100B lemmas...

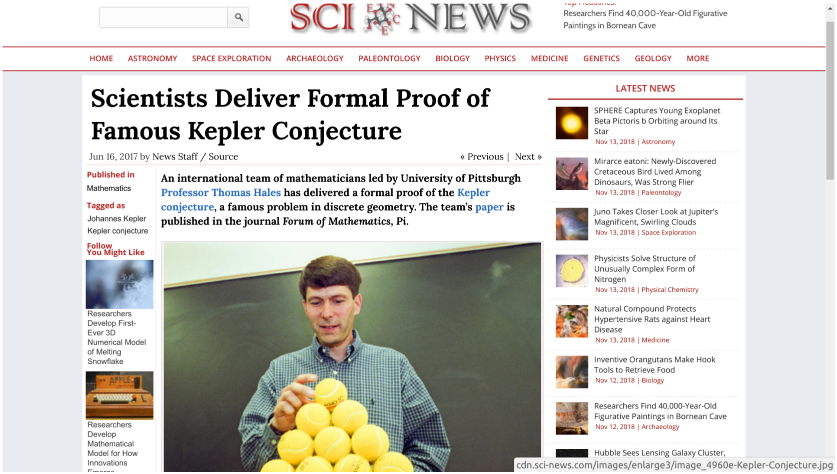
Not So Distant Future


AI4REASON
Indeed, it is similar to a less known problem B number 13501 in my knowledge base. We can use a similar polynomial reduction to planar graphs as in B, and for the resulting constraint-solving problem we use a modified version Y of the O(n⁹) algorithm X published last year in Proc. of Indian Conf. on Graph Theory.

Not So Distant Future


AI4REASON
Here is my verified formal proof with 100k basic inference steps. Here are two high-level versions of the proof, one for experts and one for textbooks.

Today: Computers Checking Large Math Proofs



SCI NEWS

Researchers Find 40,000-Year-Old Figurative Paintings in Bornean Cave

HOME ASTRONOMY SPACE EXPLORATION ARCHAEOLOGY PALEONTOLOGY BIOLOGY PHYSICS MEDICINE GENETICS GEOLOGY MORE

Scientists Deliver Formal Proof of Famous Kepler Conjecture

Jun 16, 2017 by News Staff / Source

« Previous | Next »

Published in
Mathematics

Tagged as
Johannes Kepler
Kepler conjecture

Follow
You Might Like

Researchers Develop First-Ever 3D Numerical Model of Melting Snowflake

Researchers Develop Mathematical Model for How Innovations

An international team of mathematicians led by University of Pittsburgh Professor Thomas Hales has delivered a formal proof of the Kepler conjecture, a famous problem in discrete geometry. The team's paper is published in the journal *Forum of Mathematics, Pi*.

LATEST NEWS

SPHERE Captures Young Exoplanet Beta Pictoris b Orbiting around Its Star
Nov 13, 2018 | Astronomy

Mirarce eatoni: Newly-Discovered Cretaceous Bird Lived Among Dinosaurs, Was Strong Flier
Nov 13, 2018 | Paleontology

Juno Takes Closer Look at Jupiter's Magnificent, Swirling Clouds
Nov 13, 2018 | Space Exploration

Physicists Solve Structure of Unusually Complex Form of Nitrogen
Nov 13, 2018 | Physical Chemistry

Natural Compound Protects Hypertensive Rats against Heart Disease
Nov 13, 2018 | Medicine

Inventive Orangutans Make Hook Tools to Retrieve Food
Nov 12, 2018 | Biology

Researchers Find 40,000-Year-Old Figurative Paintings in Bornean Cave
Nov 12, 2018 | Archaeology

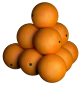
Hubble Sees Lensing Galaxy Cluster,

cdn.sci-news.com/images/enlarge3/image_4960e-Kepler-Conjecture.jpg

Kepler conjecture, Automated Reasoning and AI

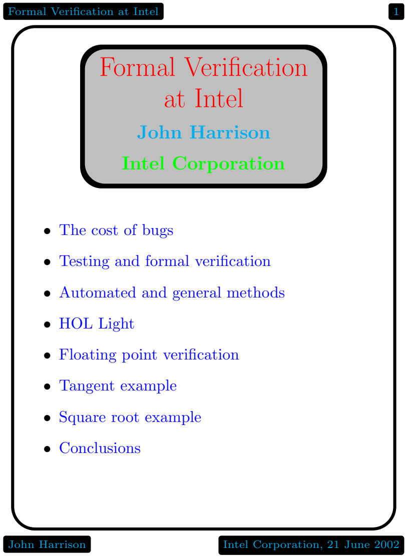
- J. Kepler (1611, Prague): The most compact way of stacking balls of the same size in space is a pyramid.

$$V = \frac{\pi}{\sqrt{18}} \approx 74\%$$



- Big proof: 300 pages + computations (Hales, Fergusson, 1998)
- Formal proof finished in 2014, 20000 theorems & proofs
- All of it computer-understandable and verified
- `polyhedron s /\ c face_of s ==> polyhedron c`
- My work:
  - Learn/reason automatically over the large corpus of proofs
  - Our methods can fully automate 40% of the proofs (2014)

Applications – Verification of HW Designs at Intel

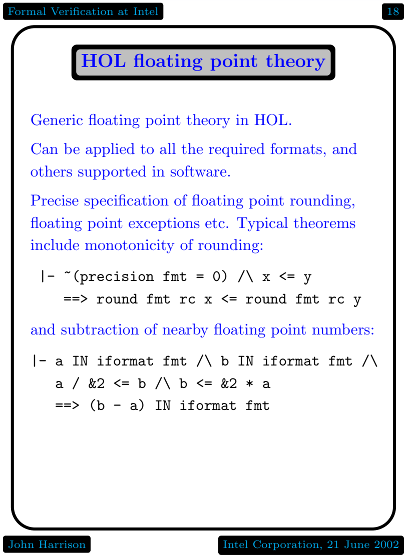


Formal Verification at Intel

John Harrison

Intel Corporation

- The cost of bugs
- Testing and formal verification
- Automated and general methods
- HOL Light
- Floating point verification
- Tangent example
- Square root example
- Conclusions



HOL floating point theory

Generic floating point theory in HOL.

Can be applied to all the required formats, and others supported in software.

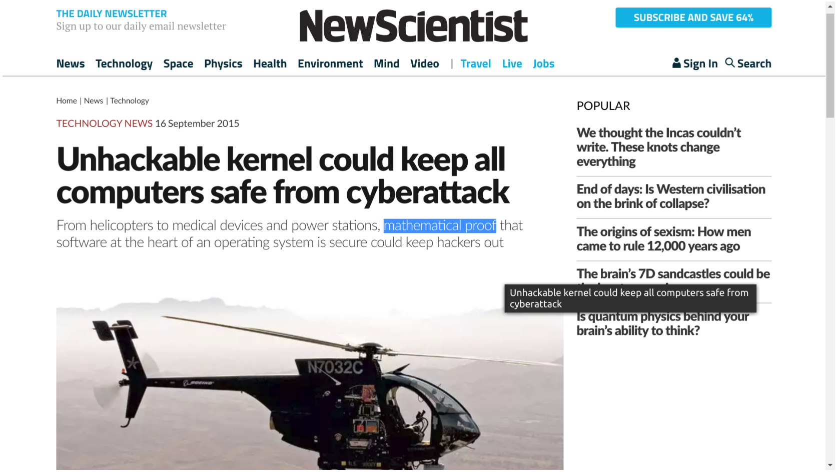
Precise specification of floating point rounding, floating point exceptions etc. Typical theorems include monotonicity of rounding:

```
 |- ~(precision fmt = 0) /\ x <= y
    ==> round fmt rc x <= round fmt rc y
```

and subtraction of nearby floating point numbers:

```
|- a IN iformat fmt /\ b IN iformat fmt /\
   a / &2 <= b /\ b <= &2 * a
   ==> (b - a) IN iformat fmt
```

Applications – Verified Operating Systems



THE DAILY NEWSLETTER
Sign up to our daily email newsletter

NewScientist

SUBSCRIBE AND SAVE 64%

News Technology Space Physics Health Environment Mind Video | Travel Live Jobs Sign In Search

Home | News | Technology

TECHNOLOGY NEWS 16 September 2015

Unhackable kernel could keep all computers safe from cyberattack

From helicopters to medical devices and power stations, mathematical proof that software at the heart of an operating system is secure could keep hackers out

POPULAR

We thought the Incas couldn't write. These knots change everything

End of days: Is Western civilisation on the brink of collapse?

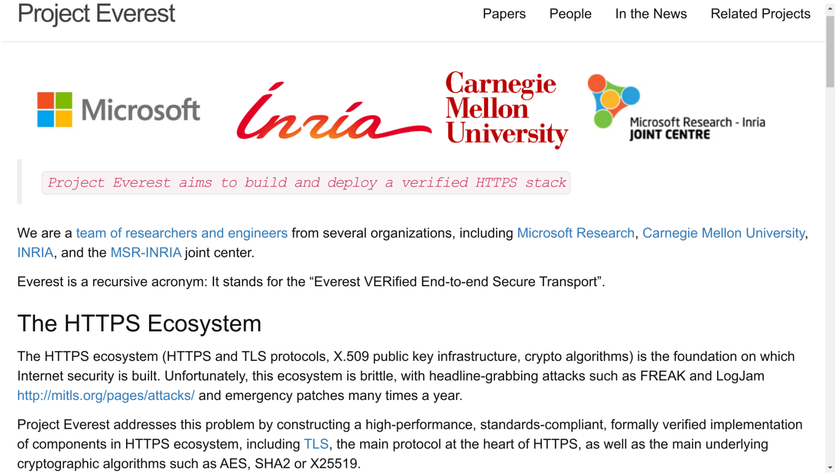
The origins of sexism: How men came to rule 12,000 years ago

The brain's 7D sandcastles could be

Unhackable kernel could keep all computers safe from cyberattack

Is quantum physics behind your brain's ability to think?

Applications – Verified Internet Protocols



Project Everest

Papers People In the News Related Projects

Microsoft Inria Carnegie Mellon University Microsoft Research - Inria JOINT CENTRE

> `Project Everest aims to build and deploy a verified HTTPS stack`

We are a team of researchers and engineers from several organizations, including Microsoft Research, Carnegie Mellon University, INRIA, and the MSR-INRIA joint center.

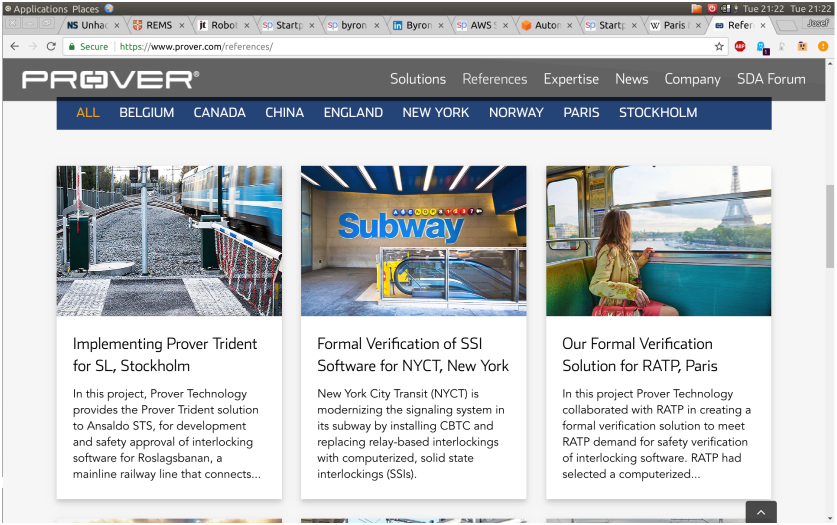
Everest is a recursive acronym: It stands for the “Everest VERified End-to-end Secure Transport”.

The HTTPS Ecosystem

The HTTPS ecosystem (HTTPS and TLS protocols, X.509 public key infrastructure, crypto algorithms) is the foundation on which Internet security is built. Unfortunately, this ecosystem is brittle, with headline-grabbing attacks such as FREAK and LogJam http://mitls.org/pages/attacks/ and emergency patches many times a year.

Project Everest addresses this problem by constructing a high-performance, standards-compliant, formally verified implementation of components in HTTPS ecosystem, including TLS, the main protocol at the heart of HTTPS, as well as the main underlying cryptographic algorithms such as AES, SHA2 or X25519.

Applications – Verified Transport Systems



Implementing Prover Trident for SL, Stockholm

In this project, Prover Technology provides the Prover Trident solution to Ansaldo STS, for development and safety approval of interlocking software for Roslagsbanan, a mainline railway line that connects...

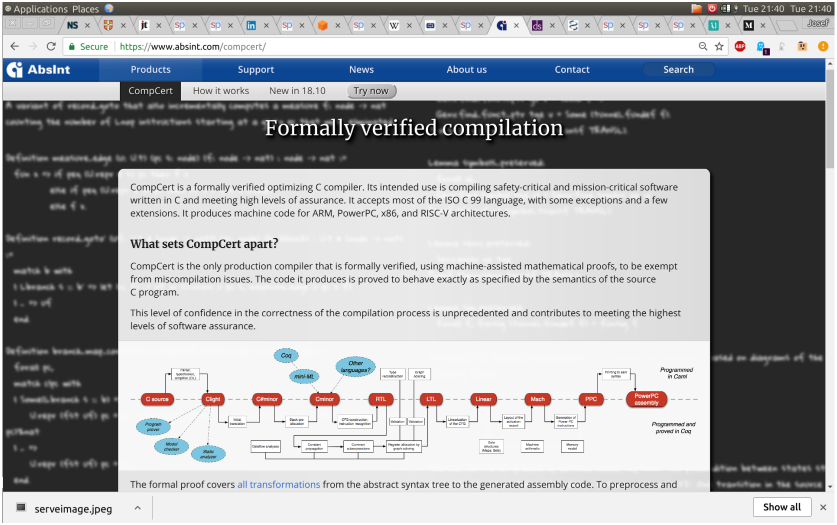
Formal Verification of SSI Software for NYCT, New York

New York City Transit (NYCT) is modernizing the signaling system in its subway by installing CBTC and replacing relay-based interlockings with computerized, solid state interlockings (SSIs).

Our Formal Verification Solution for RATP, Paris

In this project Prover Technology collaborated with RATP in creating a formal verification solution to meet RATP demand for safety verification of interlocking software. RATP had selected a computerized...

Applications – Verified Compilers



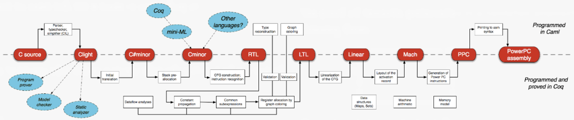
Formally verified compilation

CompCert is a formally verified optimizing C compiler. Its intended use is compiling safety-critical and mission-critical software written in C and meeting high levels of assurance. It accepts most of the ISO C 99 language, with some exceptions and a few extensions. It produces machine code for ARM, PowerPC, x86, and RISC-V architectures.

What sets CompCert apart?

CompCert is the only production compiler that is formally verified, using machine-assisted mathematical proofs, to be exempt from miscompilation issues. The code it produces is proved to behave exactly as specified by the semantics of the source C program.

This level of confidence in the correctness of the compilation process is unprecedented and contributes to meeting the highest levels of software assurance.



The formal proof covers all transformations from the abstract syntax tree to the generated assembly code. To preprocess and

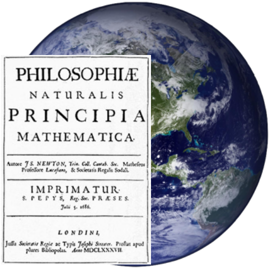
Minimal Example – Proving Equivalence of Two Programs

```
(* simple list reversal - runs in quadratic time *)
primrec rev :: "'a list => 'a list" where
"rev [] = []" |
"rev (x # xs) = rev xs @ [x]"

(* more advanced list reversal - runs in linear time *)
primrec itrev:: "'a list => 'a list => 'a list" where
  "itrev [] ys = ys" |
  "itrev (x#xs) ys = itrev xs (x#ys)"

strategy CDInd=Thens [Conjecture,Fastforce,Quickcheck,DInd]
strategy DInd_Or_CDInd = Ors [DInd, CDInd]

lemma "itrev xs [] = rev xs"
find_proof DInd_Or_CDInd
apply (subgoal_tac "\forall y. itrev xs y = Demo.rev xs @ y")
apply fastforce
apply (induct xs)
apply auto
done
```

The Technology: Automated Theorem Provers

Theorem Proving: Big Picture

Real World Problem



PHILOSOPHIÆ
NATURALIS
PRINCIPIA
MATHEMATICA.

Autore *J S. NEWTON, Trin. Coll. Cantab. Soc.* Matheſeos Profeſſore *Lucaſiano*, & Societatis Regalis Sodali.

IMPRIMATUR.
S. PEPYS, *Reg. Soc.* PRÆSES.
Julii 5. 1686.

LONDINI,
Juſſu *Societatis Regiæ* ac Typis *Joſephi Streater*. Proſtat apud plures Bibliopolas. *Anno* MDCLXXXVII.

Formalized Problem

$$X : human(X) \quad mortal(X)$$
$$X : philosopher(X) \quad human(X)$$
$$philosopher(socrates)$$
$$\overset{?}{\models}$$
$$mortal(socrates)$$

ATP

Proof Search

Proof

or

Countermodel

or

Timeout

Simple Theorem Prover: leanCoP

- lean Connection Prover – building *proof trees*
- gets first-order *clauses*, *extension* and *reduction* steps
- proof finished when all branches of the tree are *closed*
- a lot of nondeterminism, requires backtracking
- the search space quickly explodes

Clauses:

$c_1 : P(x)$

$c_2 : R(x,y) \vee \neg P(x) \vee Q(y)$

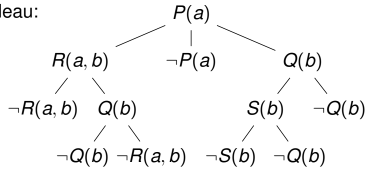
$c_3 : S(x) \vee \neg Q(b)$

$c_4 : \neg S(x) \vee \neg Q(x)$

$c_5 : \neg Q(x) \vee \neg R(a,x)$

$c_6 : \neg R(a,x) \vee Q(x)$

Closed Connection Tableau:



Using Reinforcement Learning to Guide leanCoP

- Monte-Carlo Tree Search (MCTS) – used in AlphaGo
- MCTS search nodes are sequences of clause application
- a good heuristic to explore new vs exploit good nodes:

$$\frac{w_i}{n_i} + c \cdot p_i \cdot \sqrt{\frac{\ln N}{n_i}} \qquad \text{(UCT - Kocsis, Szepesvari 2006)}$$

- we learn the *policy* – clause selection
- ... and the *value* – proof state evaluation
- big issue: representing clauses and proofs for learning
- many approaches - none too good yet
- deep learning far from good – we need *deep semantics*
- feedback loop between proving and learning - many iterations

Statistical Guidance of Connection Tableau – rlCoP

- On 32k Mizar40 problems using 200k inference limit
- nonlearning CoPs:

System	leanCoP	bare prover	rlCoP no policy/value
Training problems proved	10438	4184	7348
Testing problems proved	**1143**	431	804
Total problems proved	11581	4615	8152

- rlCoP with policy/value after 5 proving/learning iters on the training data
- $1624/1143 = 42.1\%$ improvement over leanCoP on the testing problems

Iteration	1	2	3	4	5	6	7	8
Training proved	12325	13749	14155	14363	14403	14431	14342	**14498**
Testing proved	1354	1519	1566	1595	**1624**	1586	1582	1591

Future Potential - Science

- Use strong AI/reasoning and formal verification for:
- Science
  - Routinely verify complex math, software, hardware?
  - Make all of math/science computer-understandable?
  - Strong AI assistants for math/science?
- Examples
  - Automatically understand/verify/explain all arXiv papers?
  - Can we train a superhuman system like AlphaGo/Zero for math/physics? What will it take?
  - Can we prove that the Amazon Cloud cannot be hacked?
  - The same for critical government/private IT systems?

Future Potential - Society

- Use strong AI/reasoning and formal verification for:
- Society
  - Leibniz's dream: Let us Calculate! (solve any dispute)
  - J. McCarthy: Mathem. Objectivity and the Power of Initiative
  - AI/reasoning assistants for law/regulations
  - Verification of financial, transport/traffic systems, ...
  - Explainable and very securely verified systems
- Examples
  - Prove that two Paris metro trains will never crash?
  - Prove that a trading system doesn't violate regulations?
  - Prove that a new law is inconsistent with an old one?
  - Automatically debunk fallacies in political campaigns?

Links and Impacts on Other AI Areas

- Main areas: Machine Learning, Automated Reasoning
- Needs advances in Representation Learning
- AI needs intuition, but also reasoning and explanations
- Impact on Formal Verification (SW, HW, etc.)
- Potentially on any (hard) science/thinking/arguing
- Alan Turing, 1950, AI:

“We may hope that machines will eventually compete with men in all purely intellectual fields.”

Outlook – Scientific Revolution, AI?

- What did Kepler, Galileo & Co start to do in 1600s?
- What are we trying to do today?
- Kepler’s Conjecture in Strena in 1611 (with many others)
- Kepler’s laws, Newton, ..., age of science, math, machines
- ..., Hilbert, ..., Turing, ... age of computing machines?
- 1998 machine helps to find a proof of Kepler’s Conjecture
- 2014 machine verifies a proof of Kepler’s Conjecture
- ... 2050? machine finds a proof of Kepler’s Conjecture?