Kepler and Hales: Conjectures and Proofs Dreams and Their Realization

Josef Urban

Czech Technical University in Prague

Pittsburgh, June 21, 2018

European Research Council
Established by the European Commission

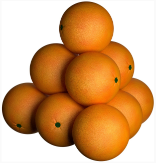
This talk is an experiment

- Normally I give “serious” talks – I am considered crazy enough
- But this is Tom’s Birthday Party, so we are allowed to have some fun:
- *This could be in principle a talk about our work on automation and AI for reasoning and formalization, and Tom’s great role in these areas. But the motivations and allusions go back to alchemistic Prague of 1600s and the (un)scientific pursuits of then versions of "singularity", provoking comparisons with our today’s funny attempts at building AI and reasoning systems for solving the great questions of Math, the Universe and Everything. I wonder where this will all take us.*
- Part 1 (Proofs?):
  - Learning automated theorem proving on top of Flyspeck and other corpora
  - Learning formalization (*autoformalization*) on top of them
- Part 2 (Conjectures?):
  - How did we get here?
  - What were Kepler & Co trying to do in 1600s?
  - What are we trying to do today?

The Flyspeck project – A Large Proof Corpus

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

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



- Proved by Hales & Ferguson in 1998, 300-page proof + computations
- Big: Annals of Mathematics gave up reviewing after 4 years
- Formal proof finished in 2014
- 20000 lemmas in geometry, analysis, graph theory
- All of it at `https://code.google.com/p/flyspeck/`
- All of it computer-understandable and verified in HOL Light:
- `polyhedron s /\ c face_of s ==> polyhedron c`
- However, this took 20 – 30 person-years!

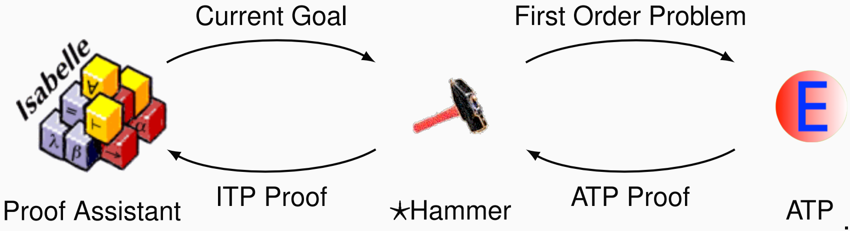
Automated Theorem Proving (ATP) in Large Theories

- Can we automatically prove simple conjectures over large theories like Flyspeck?
- In standard ATP we start with a few axioms and symbols
- Standard ATPs human-optimized for small theories (groups)
- In large-theory ATP we start with thousands of axioms and symbols
- The combinatorial explosion of ATP shows from the very beginning
- Algorithms for fact selection and proof guidance developed recently
- Such algorithms often use machine learning methods
- Use large corpora like Flyspeck for learning math thinking?
- Combine learning and reasoning? Strong AI?

Early Machine Learning for Fact Selection over Mizar

- 2003: Can existing ATPs (E, SPASS, Vampire) be used on the freshly translated Mizar library?
- About 80000 nontrivial math facts at that time – impossible to use them all
- Mizar Proof Advisor (2003):
- Learn fact selection from previous proof!
- Recommend relevant premises when proving new conjectures
- Give them to existing ATPs
- First results over the whole Mizar library in 2003:
  - about 70% coverage in the first 100 recommended premises
  - chain the recommendations with strong ATPs to get full proofs
  - about 14% of the Mizar theorems were then automatically provable (SPASS)
  - sometimes we can find simpler proofs!
- Done with much more developed tools for Flyspeck in 2012

Today’s AI-ATP systems (⋆-Hammers)



Current Goal

First Order Problem

ITP Proof

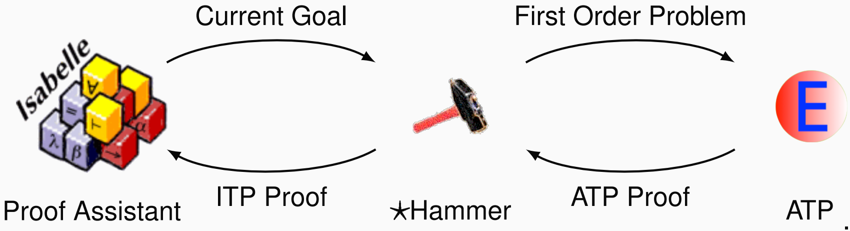
ATP Proof

Proof Assistant

⋆Hammer

ATP .

Today’s AI-ATP systems (⋆-Hammers)



Current Goal

First Order Problem

ITP Proof

ATP Proof

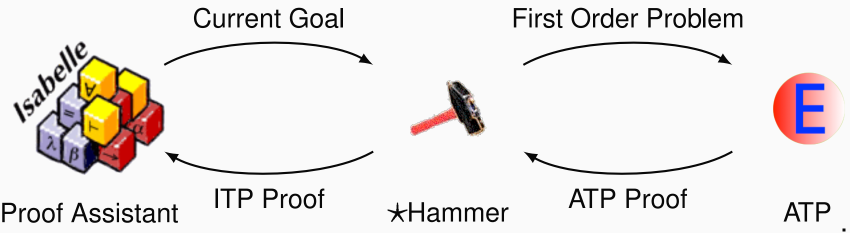
Proof Assistant

⋆Hammer

ATP

How much can it do?

Today's AI-ATP systems (⋆-Hammers)



Current Goal

First Order Problem

ITP Proof

ATP Proof

Proof Assistant

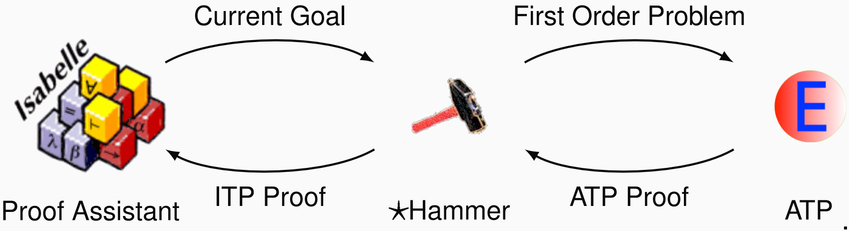
⋆Hammer

ATP

How much can it do?

- Mizar / MML – MizAR
- Isabelle (Auth, Jinja) – Sledgehammer (Paulson, Blanchette)
- Flyspeck (including core HOL Light and Multivariate) – HOL(y)Hammer
- HOL4 (Gauthier and Kaliszyk)
- CoqHammer (Czajka and Kaliszyk) - about 40% on Coq standard library

Today's AI-ATP systems (⋆-Hammers)



Current Goal

First Order Problem

ITP Proof

ATP Proof

Proof Assistant

⋆Hammer

ATP .

How much can it do?

- Mizar / MML – MizAR
- Isabelle (Auth, Jinja) – Sledgehammer (Paulson, Blanchette)
- Flyspeck (including core HOL Light and Multivariate) – HOL(y)Hammer
- HOL4 (Gauthier and Kaliszyk)
- CoqHammer (Czajka and Kaliszyk) - about 40% on Coq standard library

$\approx$ 45% success rate

FACE_OF_POLYHEDRON_POLYHEDRON

```
let FACE_OF_POLYHEDRON_POLYHEDRON = prove
 (`!s:real^N->bool c. polyhedron s /\ c face_of s ==> polyhedron c`,
  REPEAT STRIP_TAC THEN FIRST_ASSUM
   (MP_TAC o GEN_REWRITE_RULE I [POLYHEDRON_INTER_AFFINE_MINIMAL]) THEN
  REWRITE_TAC[RIGHT_IMP_EXISTS_THM; SKOLEM_THM] THEN
  SIMP_TAC[LEFT_IMP_EXISTS_THM; RIGHT_AND_EXISTS_THM; LEFT_AND_EXISTS_THM] THEN
  MAP_EVERY X_GEN_TAC
   [`f:(real^N->bool)->bool`; `a:(real^N->bool)->real^N`;
    `b:(real^N->bool)->real`] THEN
  STRIP_TAC THEN
  MP_TAC(ISPECL [`s:real^N->bool`; `f:(real^N->bool)->bool`;
                 `a:(real^N->bool)->real^N`; `b:(real^N->bool)->real`]
         FACE_OF_POLYHEDRON_EXPLICIT) THEN
  ANTS_TAC THENL [ASM_REWRITE_TAC[] THEN ASM_MESON_TAC[]; ALL_TAC] THEN
  DISCH_THEN(MP_TAC o SPEC `c:real^N->bool`) THEN ASM_REWRITE_TAC[] THEN
  ASM_CASES_TAC `c:real^N->bool = {}` THEN
  ASM_REWRITE_TAC[POLYHEDRON_EMPTY] THEN
  ASM_CASES_TAC `c:real^N->bool = s` THEN ASM_REWRITE_TAC[] THEN
  DISCH_THEN SUBST1_TAC THEN MATCH_MP_TAC POLYHEDRON_INTERS THEN
  REWRITE_TAC[FORALL_IN_GSPEC] THEN
  ONCE_REWRITE_TAC[SIMPLE_IMAGE_GEN] THEN
  ASM_SIMP_TAC[FINITE_IMAGE; FINITE_RESTRICT] THEN
  REPEAT STRIP_TAC THEN REWRITE_TAC[IMAGE_ID] THEN
  MATCH_MP_TAC POLYHEDRON_INTER THEN
  ASM_REWRITE_TAC[POLYHEDRON_HYPERPLANE]);;
```

FACE_OF_POLYHEDRON_POLYHEDRON

```
polyhedron s /\ c face_of s ==> polyhedron c
```

- HOL Light proof: could not be re-played by ATPs
- Alternative proof found by a hammer based on `FACE_OF_STILLCONVEX`
- Face *t* of a convex set *s* is equal to the intersection of *s* with the affine hull of *t*.
- AI selection of relevant facts can today sometimes beat humans!

```
FACE_OF_STILLCONVEX:
 !s t:real^N->bool. convex s ==>
 (t face_of s <=>
  t SUBSET s /\ convex(s DIFF t) /\ t = (affine hull t) INTER s)
POLYHEDRON_IMP_CONVEX:
 !s:real^N->bool. polyhedron s ==> convex s
POLYHEDRON_INTER:
 !s t:real^N->bool. polyhedron s /\ polyhedron t
   ==> polyhedron (s INTER t)
POLYHEDRON_AFFINE_HULL:
 !s. polyhedron(affine hull s)
```

Using Learning to Guide Theorem Proving

- **high-level**: pre-select lemmas from a large library, give them to ATPs
- **high-level**: pre-select a good ATP strategy/portfolio for a problem
- **high-level**: pre-select good *hints* for a problem, use them to guide ATPs
- **low-level**: guide every inference step of ATPs (tableau, superposition)
- **low-level**: guide every kernel step of LCF-style ITPs
- **mid-level**: guide application of tactics in ITPs
- **mid-level**: invent suitable ATP strategies for classes of problems
- **mid-level**: invent suitable conjectures for a problem
- **mid-level**: invent suitable concepts/models for problems/theories
- **proof sketches**: explore stronger/related theories to get proof ideas
- **theory exploration**: develop interesting theories by conjecturing/proving
- **feedback loops**: (dis)prove, learn from it, (dis)prove more, learn more, ...
- ...

Statistical Guidance of Connection Tableau

- learn guidance of every clausal inference in connection tableau (leanCoP)
- set of first-order clauses, *extension* and *reduction* steps
- proof finished when all branches are closed
- a lot of nondeterminism, requires backtracking
- *Iterative deepening* used in leanCoP to ensure completeness
- good for learning – the tableau compactly represents the proof state

Clauses:

$c_1 : P(x)$

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

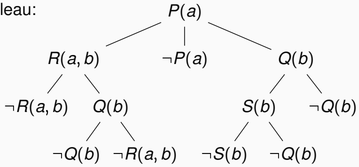
$c_3 : S(x) \lor \neg Q(b)$

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

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

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

Closed Connection Tableau:



Statistical Guidance of Connection Tableau – rlCoP

- **MaLeCoP** (2011): first prototype Machine Learning Connection Prover
- Fairly Efficient MaLeCoP = **FEMaLeCoP** (15% better than leanCoP)
- 2018: stronger learners via C interface to OCAML (boosted trees)
- remove iterative deepening, the prover can go deep (completeness bad!)
- Monte-Carlo Tree Search (MCTS) governs the search (AlphaGo/Zero!)
- MCTS search nodes are sequences of clause application
- a good heuristic to explore new vs exploit good nodes:

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

- learning both *policy* (clause selection) and *value* (state evaluation)
- clauses represented not by names but also by features (generalize!)
- binary learning setting used: | proof state | clause features |
- mostly term walks of length 3 (trigrams), hashed into small integers
- many iterations of proving and learning

Statistical Guidance of Connection Tableau – rlCoP

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

System	leanCoP	bare prover	rlCoP no policy/value (UCT only)
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

TacticToe: Tactic Guidance of ITPs (Gauthier et al.)

- learns from human tactical HOL4 proofs to solve new goals
- no translation or reconstruction needed
- similar to rlCoP: policy/value learning
- however much more technically challenging:
    - tactic and goal state recording
    - tactic argument abstraction
    - absolutization of tactic names
    - nontrivial evaluation issues
- policy: which tactic/parameters to choose for a current goal?
- value: how likely is this proof state succeed?
- 2018: 66% of HOL4 toplevel proofs in 60s (better than a hammer!)
- work in progress for Coq

Statistical/Semantic Parsing of Informalized Flyspeck

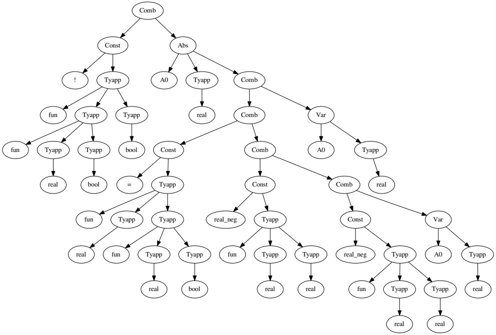
- Goal: Learn understanding of informal math formulas and reasoning
- Statistical translation combined with semantic hammers in feedback loops
- Training/testing examples exported from Flyspeck formulas (thanks Tom!)
- Along with their informalized versions
- First version: Train probabilistic parser on 20k grammar parse trees
  - Annotate each (nonterminal) symbol with its HOL type
  - Also “semantic (formal)” nonterminals annotate overloaded terminals
  - guiding analogy: word-sense disambiguation using CYK is common
- Terminals exactly compose the textual form, for example:
- REAL_NEGNEG: $\forall x. - -x = x$

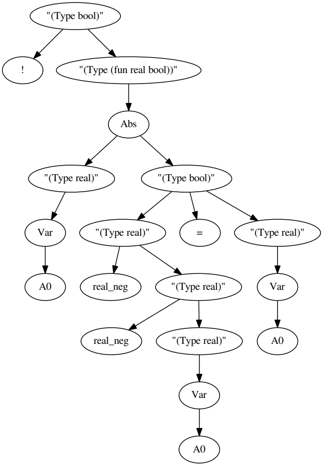
```
(Comb (Const "!" (Tyapp "fun" (Tyapp "fun" (Tyapp "real") (Tyapp "bool"))
(Tyapp "bool"))) (Abs "A0" (Tyapp "real") (Comb (Comb (Const "=" (Tyapp "fun"
(Tyapp "real") (Tyapp "fun" (Tyapp "real") (Tyapp "bool")))) (Comb (Const
"real_neg" (Tyapp "fun" (Tyapp "real") (Tyapp "real"))) (Comb (Const
"real_neg" (Tyapp "fun" (Tyapp "real") (Tyapp "real"))) (Var "A0" (Tyapp
"real"))))) (Var "A0" (Tyapp "real")))))
```

- becomes

```
("¨(Type bool)"̈ ! ("¨(Type (fun real bool))"̈ (Abs ("¨(Type real)"̈
(Var A0)) ("¨(Type bool)"̈ ("¨(Type real)"̈ real_neg ("¨(Type real)"̈
real_neg ("¨(Type real)"̈ (Var A0)))) = ("¨(Type real)"̈ (Var A0))))))
```

Example grammar trees



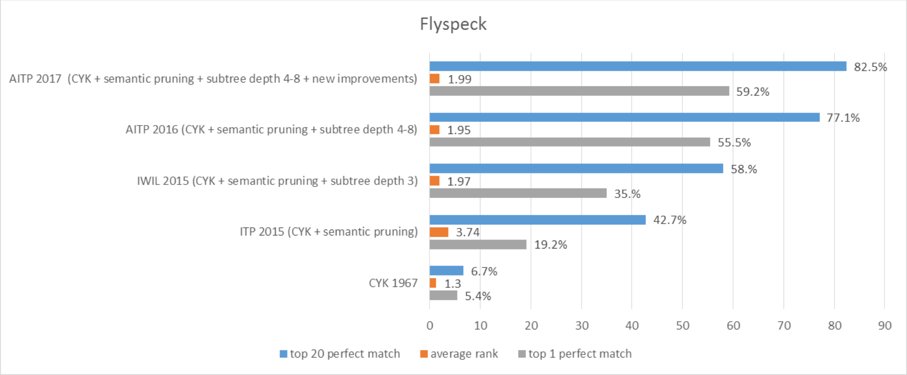


Online parsing system

- Induce PCFG (probabilistic context-free grammar) from the trees
- CYK: dynamic-programming algorithm for parsing ambiguous sentences
- input: sentence – a sequence of words and a binarized PCFG
- output: N most probable parse trees
- Transformed to HOL parse trees (preterms, Hindley-Milner)
- typed checked in HOL and then given to an ATP (hammer)
- `“sin ( 0 * x ) = cos pi / 2”`
- produces 16 parses
- of which 11 get type-checked by HOL Light as follows
- with all but three being proved by HOL(y)Hammer

```
sin (&0 * A0) = cos (pi / &2) where A0:real
sin (&0 * A0) = cos pi / &2 where A0:real
sin (&0 * &A0) = cos (pi / &2) where A0:num
sin (&0 * &A0) = cos pi / &2 where A0:num
sin (&(0 * A0)) = cos (pi / &2) where A0:num
sin (&(0 * A0)) = cos pi / &2 where A0:num
csin (Cx (&0 * A0)) = ccos (Cx (pi / &2)) where A0:real
csin (Cx (&0) * A0) = ccos (Cx (pi / &2)) where A0:real^2
Cx (sin (&0 * A0)) = ccos (Cx (pi / &2)) where A0:real
csin (Cx (&0 * A0)) = Cx (cos (pi / &2)) where A0:real
csin (Cx (&0) * A0) = Cx (cos (pi / &2)) where A0:real^2
```

Flyspeck Progress



Neural Autoformalization (Wang et al., 2018)

- generate about 1M Latex - Mizar pairs based on Bancerek's work
- train neural seq-to-seq translation models (Luong – NMT)
- evaluate on about 100k examples
- many architectures tested, some work much better than others
- very important latest invention: *attention* in the seq-to-seq models
- more data very important for neural training – our biggest bottleneck
- You can help! – Just contribute to Tom's Formal Abstracts project!
- Many example from yesterday's talks: *Integers act (anti)holomorphically*?
- AI feedback loops:
- more disambiguated formal math, more formal proofs, stronger automated reasoning via learning, stronger semantic filtering helping the disambiguation, more/better formal data, ...

Neural Autoformalization data

Rendered LaTeX	If $X \subseteq Y \subseteq Z$, then $X \subseteq Z$.
Mizar	`X c= Y & Y c= Z implies X c= Z;`
Tokenized Mizar	`X c= Y & Y c= Z implies X c= Z ;`
LaTeX	`If $X \subseteq Y \subseteq Z$, then $X \subseteq Z$.`
Tokenized LaTeX	`If $ X \subseteq Y \subseteq Z $ , then $ X \subseteq Z $ .`

Neural Autoformalization Results

Parameter	Final Test Perplexity	Final Test BLEU	Identical Statements (%)	Identical No-overlap (%)
128 Units	3.06	41.1	40121 (38.12%)	6458 (13.43%)
256 Units	1.59	64.2	63433 (60.27%)	19685 (40.92%)
512 Units	1.6	**67.9**	66361 (63.05%)	21506 (44.71%)
1024 Units	**1.51**	61.6	**69179 (65.73%)**	**22978 (47.77%)**
2048 Units	2.02	60	59637 (56.66%)	16284 (33.85%)

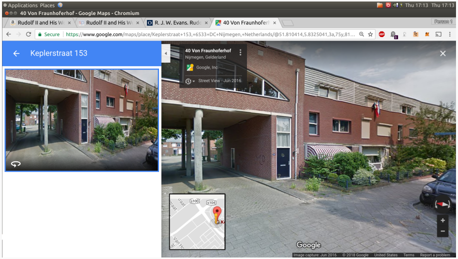
Neural Fun – Performance after Some Training

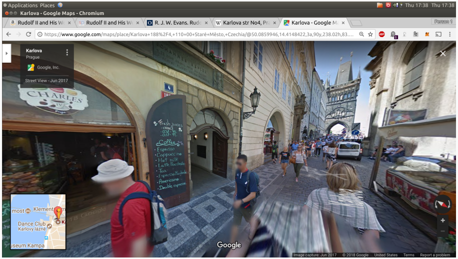
Rendered LaTeX	Suppose s_8 is convergent and s_7 is convergent . Then $\lim(s_8+s_7) = \lim s_8 + \lim s_7$
Input LaTeX	`Suppose $ { s _ { 8 } } $ is convergent and $ { s _ { 7 } } $ is convergent . Then $ \mathop { \rm lim } ( { s _ { 8 } } { + } { s _ { 7 } } } ) \mathrel { = } \mathop { \rm lim } { s _ { 8 } } { + } \mathop { \rm lim } { s  _ { 7 } } $ .`
Correct	`seq1 is convergent & seq2 is convergent implies lim ( seq1 + seq2 ) = ( lim seq1 ) + ( lim seq2 ) ;`
Snapshot-1000	`x in dom f implies ( x * y ) * ( f | ( x | ( y | ( y | y ) ) ) ) = ( x | ( y | ( y | ( y | y ) ) ) ) ) ;`
Snapshot-2000	`seq is summable implies seq is summable ;`
Snapshot-3000	`seq is convergent & lim seq = 0c implies seq = seq ;`
Snapshot-4000	`seq is convergent & lim seq = lim seq implies seq1 + seq2 is convergent ;`
Snapshot-5000	`seq1 is convergent & lim seq2 = lim seq2 implies lim_inf seq1 = lim_inf seq2 ;`
Snapshot-6000	`seq is convergent & lim seq = lim seq implies seq1 + seq2 is convergent ;`
Snapshot-7000	`seq is convergent & seq9 is convergent implies lim ( seq + seq9 ) = ( lim seq ) + ( lim seq9 ) ;`

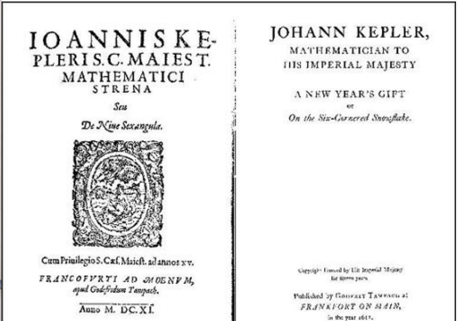
Part 2 – How did we get here?

- How did we get here?
- What were Kepler & Co trying to do in 1600s?
- What are we trying to do today?
- Kepler’s Conjecture in Strena in 1611 (among many other conjectures)
- Kepler’s laws, Galileo, Newton, ..., age of science, math, machines
- ..., Poincare, Russell, Von Neumann, Turing, ... age of computing machines?
- 1998 computing machine helps to find a proof of Kepler’s Conjecture
- 2014 computing machine verifies a proof of Kepler’s Conjecture
- ... 2050? computing machine finds a proof of Kepler’s Conjecture?

Part 2 – Kepler & Co.







IOANNIS KEPLERI S.C. MAIEST. MATHEMATICI STRENA

Seu

De Niue Sexangula.

Cum Priuilegio S. Cæſ. Maieſt. ad annos xv.

FRANCOFVRTI AD MOENVM,
apud Godefridum Tampach.

Anno M. DC. XI.

JOHANN KEPLER,
MATHEMATICIAN TO
HIS IMPERIAL MAJESTY

A NEW YEAR'S GIFT
or
On the Six-Cornered Snowflake.

Copyright Granted by His Imperial Majesty
for fifteen years

Published by GODFREY TAMPACH at
FRANKFORT ON MAIN,
in the year 1611.

What were Kepler & Co. Trying To Do?

- Kepler: Mysterium Cosmographicum (1596)
- God's geometrical plan for the universe (Platonic solids)
- Didn't work – he needed data!
- Off he went to Prague in 1600 - Tycho Brahe
- Brahe was collecting precise data since the supernova of 1572
- Kicked out off Denmark and invited by Rudolf II in 1599
- Imperial Court Astronomer
- Kepler appointed the imperial mathematician after Brahe's death
- Observed snowflakes while walking across the Charles bridge in 1610

Prague around 1600s

- Rudolf II inviting all sorts artists, alchemist, astrologists (eventually astronomers, geometers, chemists, scientists?)
- Brahe, Dee, Kelly, Kepler, Sendivogius, Rabbi Loew ben Bezalel, ...
- About 40 alchemists at the top - working hard on:
  - searching for the Philosopher's Stone (gold, immortality, and all that)
  - cosmic harmony through numbers and combinations of symbols
  - mysticism, Kabbalah and "Harmony of spheres"
- Dee's claims that world will soon go through a miracle reform
- Kepler's claim that Earth has a soul – subjected to astrological harmony
- Conjecturing about the shape of snowflakes and stacking of spheres?
- What gives the shell of the snail its spiral form?
- Why do most flowers have five petals?
- Explain the Universe through math and geometry? Scientific Revolution?

Scientific Revolution of 1600s: Data, Math, Induction

- Brahe, Kepler, Galileo, Newton, Bacon, Descartes, ..
- Experience Not Only Doctrine! Data-driven science! (*Natural Philosophy*)
- Bacon: not just Aristotelian deduction/syllogisms, but *induction* from data
- *Men have sought to make a world from their own conception and to draw from their own minds all the material which they employed, but if, instead of doing so, they had consulted experience and observation, they would have the facts and not opinions to reason about, and might have ultimately arrived at the knowledge of the laws which govern the material world.*
- Mathematization of philosophy: rebellion by the mathematicians against philosophers
- apparent in Newton's title: Mathematical Principles of Natural Philosophy
- Before Newton there was Kepler:
- Kepler's New Astronomy deliberately conflates the worlds of the mathematician (who deals with astronomy) and natural philosopher (who deals with physics and with causation in nature)

Their Science Revolution vs AI Revolution of Today?

- They were obsessed with the nature, its data and conjectures over them
- Produced lots of *observation/experimental data*
- And we are obsessed with our thinking process (AI)
- Pat Langley (PhD at CMU under H. Simon) – the Bacon system (1978):
- *learn the laws*: Kepler’s third law , Coulomb’s law, Ohm’s law, ...
- ... from the observation data
- But they also produced *thinking data* (science and math)
- Today the thinking data is made understandable to our *thinking machines*
- Today’s proof libraries are our data, and we are trying to come up with algorithms that *learn our thinking*
- Shall we teach machines to discover the proof of Kepler’s conjecture?

Their Science Revolution vs AI Revolution of Today?

- They strive to understand and predict the universe, explore and exploit it
- Among other things leading to the discovery of many useful machines
- Among them the *computing machine*
- We strive to understand and speed up our discovery process
- by better and better algorithms for the computing machines
- Development of thinking machines? (singularity?)
- I wonder what would Kepler say to today's crazy Simulation conjectures linking God(s) with builders of thinking machines and simulations?

Where does this lead us?

- Acceleration of science? - feedback loops between:
- Better observation/exploration, more data, better theories, better thinking tools, even better theories, even better observations, ...
- Is this already some "singularity"?
- Not clear - e.g. in AlphaGo/Zero the feedback loop plateaus at some point
- In a similar way our learning/proving loops currently plateau
- There might be physical limits to our observation and thinking tools limiting how far the feedback loops can get
- But physics as we know it will likely be quickly outdated when automated science takes off
- ... and likely also our current thinking tools
- It’s very hard to believe that our (or our thinking successors’) discovery process of the universe could stop somehow
- At least math seems at the moment unfinishable

The End: Dreams and their Realizations

- I'll stop here
- My dreaming/conjecturing is clearly weak compared to Kepler's
- Just read his Dream (Somnium): First sci-fi (1608) – flying to the Moon
- He almost predicts rockets:
- *The initial shock is the worst part, since they are launched upward as if by a gunpowder explosion and they fly over mountains and seas.*
- The man who realizes the dream is the Icelander Duracotus
- Tom, you come from Iceland, and you are realizing Kepler's dreams
- So I wish you many more flights to many interesting places!