Automating and Presenting Formal Proofs and What Could We Do for Formal Physics

Cezary Kaliszyk — University of Innsbruck

Josef Urban — Radboud University

Montreal, August 26, 2014

Workshop on Physics Formalization

Outline

Automation and Proof Presentation Demo for Mizar

- Calling automated theorem provers (ATPs) remotely on Mizar goals (Emacs): https://www.youtube.com/watch?v=4es4iJKtM3I
- Complete cross-linking of the formal texts for fast understanding (Emacs and HTML: http://mizar.cs.ualberta.ca/~mptp/8.1.03_5.23.1213/html/rvsum_3.html#T46
- Linking to corresponding informal descriptions in Wikipedia (this could be done for physics and HOL Light too) http://mizar.cs.ualberta.ca/~mptp/8.1.03_5.23.1213/html/hermitan.html#T45

Presenting and Explaining Coq Proofs Using Proviola

- Carst Tankink's PhD work: `http://www.cs.ru.nl/~carst/`
- B. Peirce Software Foundations course in Coq/Proviola:
- `http://cs.ru.nl/~carst/sf/Lists.html#lab58`
- Proviola: The proof state after each command is recorded and displayed in a separate HTML frame
- Useful for presenting HOL (Light) proofs too

Presenting and Cross-linking Informal and Formal Flyspeck

- The Flyspeck LaTeX book wikified and displayed using MathJaX (quite easy)
- The formal Flyspeck code HTML-ized heuristically and HTML blocks extracted for each theorem
- Simple TeX annotation macros used by Hales for cross-linking between the informal and formal text
- The informal and formal code presented side-by-side in the wiki:
- `http://mws.cs.ru.nl/agora_flyspeck/flyspeck/fly_demo/`
- This was all quite simple

Strong Proof Automation: Learning How to Use Theorems from Many Proofs

- Methods that select theorems only by their similarity to the conjecture (SInE, MePo)
- Methods that re-use and generalize (learn) the information about used theorems from previous proofs (k-nearest neighbor, naive Bayes, kernel methods, etc.)
- larger AI-style loops: first prove some theorems, then learn form such proofs and again prove some more theorems (and iterate this)
- Machine Learner for Automated Reasoning - 2013 CADE Automated System Competition (CASC) winner
- `http://www.cs.miami.edu/~tptp/CASC/24/WWWFiles/DivisionSummary1.html`

Re-using Millions of Smaller Lemmas

- Experiments over HOL Light, Flyspeck and Mizar
- Over 1B low-level lemmas in Flyspeck
- 1.5M-7M higher-level lemmas in MML and Flyspeck
- Define fast preprocessing methods to extract the most important ones:
- PageRank, recursive dependency count, recursive use count, etc.
- Use the most important lemmas together with the toplevel theorems - helps by 5-20% (needs more evaluations)
- Conjecturing: guessing the intermediate lemmas in longer proofs (we do not have the methods yet)

Automated Invention of Search Strategies for Classes of Problems

- Blind Strategymaker: invent faster strategies (sets of search parameters) by mutation on groups of easy problems
- Test the faster strategies on harder problems
- When the hard problems become easy for the new strategies, include them into the training set and use for further mutation of the strategies
- Improved the performance of the E prover on Mizar and Flyspeck by about 25% after 30 hours of training

Fine-Grained Guidance of Automated Theorem Provers

- Machine Learning Connection Prover (MaLeCoP): put the AI methods inside a tableau ATP (leanCoP)
- the learning/deduction feedback loop runs across problems and inside problems
- The more problems/branches you solve/close, the more solutions you can learn from
- The more solutions you can learn from, the more problems you solve
- already about 20-time proof search shortening on MPTP Challenge compared to leanCoP

Progress in non-learning automation in the last 10-15 years

- Mike Beeeson and Larry Wos: OTTER Proofs in Tarskian Geometry (IJCAR 2014)
- 72 proofs found automatically by Otter
- 100 by Prover9
- 125 by unaided E
- so far about 146 by ET

Combining with Computer Algebra: Mike Beeson's Weierstrass

- Weierstrass was built by combining the symbolic computation facilities of Mathpert with an inference engine, earlier built for the theorem-prover Gentzen.
- Mathpert: a program for helping students learn algebra, precalculus, and calculus (CAS but more formal)
- Can Weierstrass find proofs in number theory and calculus?
- Perhaps by adding some general-purpose meta-rules (rules governing the generation of proofs)?

Mike Beeson's Weierstrass

- Indeed it can!
- automated proof of irrationality of e, continuity of $f(x) = x^3$, $f(x) = sin(x)$, etc.
- "One wonders if that would still be the case if 200 meta-rules for various subjects had been added"
- Again, we might be able to learn such rules from many formal proofs that involve computation and use them similarly as in MaLeCoP
- Formal physics could be one of the main customers of such automation research

Auto-formalization: Informal and Semiformal Corpora in 2014

- Arxiv.org: 1M articles collected over some 20 years (not just math, majority is physics)
- Wikipedia: 25,000 math articles in 2010 - collected over 10 years only
- Semiformal and informal corpora have grown one or two orders of magnitude faster than formal ones
- We should use this energy to accelerate production of computer-understandable science (math, physics, etc.)

Attempts at auto-formalization

- Claus Zinn and others:
- manual translators from latex to formal math, failing for several reasons:
- lack of the vast background knowledge that the mathematicians use for gap-filling
- lack of decent automated reasoning methods over such vast corpora of math knowledge
- lack of translation methods that can automatically adapt to large corpora, using automated self-improvement

But this has been changing in the last decade!

- we started to have reasonably big formal corpora of common math
- we have developed reasonably strong automated reasoning methods over them
- and a large part of the latter was thanks to learning methods (40% of Mizar theorems automatically provable today)
- and we are even getting some aligned informal/formal corpora: Flyspeck, Compendium of Continuous Lattices, Feith-Thompson
- so let's use what works: statistical machine translation combined with strong learning-assisted automated reasoning over large libraries providing the common background!
- We need as much aligned formal/informal corpora for this as possible (improves the machine learning methods)
- Why not do this for physics too?

One More Inspiration by Mike Beeson for Physics Formalization

- Michael Beeson: Constructivity, Computability, and the Continuum (p. 23-25)
- " ... What follows is a simple calculation showing that distances smaller than this [Planck] length cannot exist in the usual sense; i.e., spacetime cannot be considered to be smooth at that scale. The calculation uses two fundamental equations: The uncertainty principle from quantum mechanics, and the Schwarzschild radius for the formation of a black hole, from general relativity. It is often stated that 'general relativity and quantum mechanics are not consistent' ... "
- Can we make this formal? What is this "inconsistency" formally?

LHC Security as an (Future) Application?

- `http://press.web.cern.ch/backgrounders/safety-lhc`
- "According to the well-established properties of gravity, described by Einstein's relativity, it is impossible for microscopic black holes to be produced at the LHC. There are, however, some speculative theories that predict the production of such particles at the LHC. All these theories predict that these particles would disintegrate immediately. Black holes, therefore, would have no time to start accreting matter and to cause macroscopic effects. Although theory predicts that microscopic black holes decay rapidly, even hypothetical stable black holes can be shown to be harmless by studying the consequences of their production by cosmic rays."
- Can we prove formally any of these claims?
- If we find a bug during the formalization, it could prevent destruction of the Earth (about $USD 10^{14}$ yearly)