DeepAlgebra - an outline

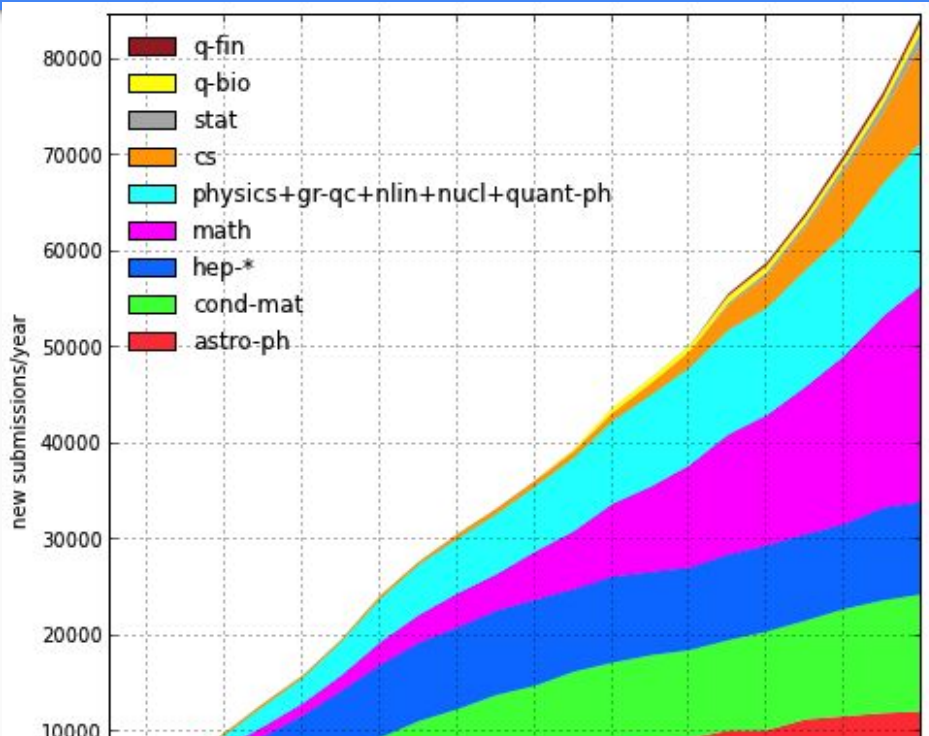
Przemyslaw Chojecki (Polish Academy of Sciences and deepsense.io BIG DATA SCIENCE)

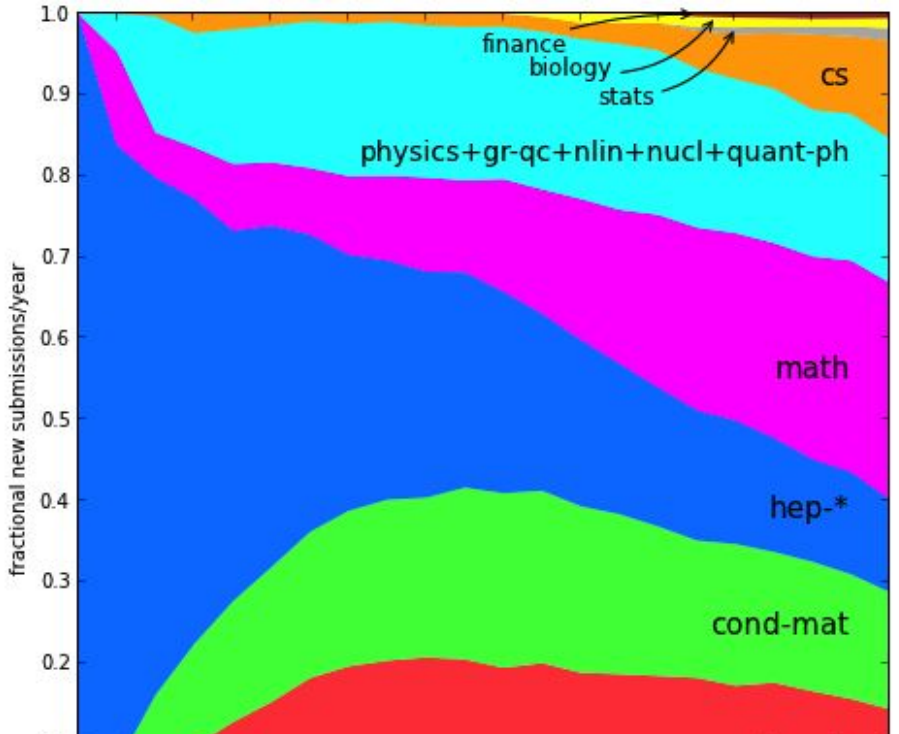
Problems within mathematics

Growing number of mathematical research (--> arXiv). More complicated, more interdependent.

Impossible to verify correctness for “outsiders” - knowledge is accepted as knowledge by a small group of experts (e.g. problem with accepting Mochizuki’s proof of abc-conjecture; not understandable to other experts).

Problems within mathematics

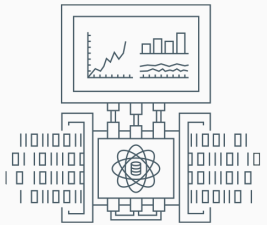




Potential solution

Automation or semi-automation of:

- Producing mathematics
- Verifying already existing mathematics



Automatic theorem proving

Current approach to automatic theorem proving:

- Take a mathematical work (e.g. Feit-Thompson theorem or proof of Kepler conjecture)
- Rewrite it in Coq/Mizar/other Interactive Theorem Prover
- Verify!

References: T. Hales, ”Developments in Formal Proofs”, Seminaire Bourbaki 1086. abs/1408.6474.

Drawbacks

1. Mathematical work is based on previous works. One needs to lay down foundation each time at least to some extent (but e.g. Mizar Math Library).
2. Tedious work of filling in gaps (human way of writing mathematics is different than what Coq/Mizar accepts).
3. Purely manual work!

Outcome

Once in Coq/Mizar, there are growing number of methods to prove new theorems:

-> hammers

-> tactics

-> machine/deep learning (?)

References: J. Blanchette, C. Kaliszyk, L. Paulson and J. Urban, ”Hammering towards QED”, J. Formalized Reasoning 9(1), pp. 101-148, doi:10.6092/issn.1972-5787/4593.

A. Alemi, F. Chollet, G. Irving, C. Szegedy, J. Urban, ”DeepMath - Deep Sequence Models for Premise Selection”, arXiv:1606.04442

Towards automation

To fully use power of machine/deep learning, one needs more data! Moreover in order to stay with current research we need to translate LaTeX -> Coq/Mizar much faster!

Need: automate translation of human-written math LaTeX work to Coq/Mizar.

NLP problem

Human-written LaTeX math file Coq/Mizar

View it as an NLP problem of creating a dictionary between two languages.

References: M. Ganesalingam “The Language of Mathematics”, LNCS 7805

Building a dictionary

Enhance usual syntactic parsers (e.g. TensorFlow's SyntaxNet) with Types and variables.

“Let G be a group” ---> “G” is a variable of Type “group”.

Use it to translate LaTeX into Coq/Mizar sentence by sentence. Still need a good source of mathematics!

Algebraic geometry

One of the pillars of modern mathematical research, quickly developing, but having a good foundation (Grothendieck's EGA/SGA, The Stacks Project).

"Abstract" hence easier to verify for computers than analytical parts of mathematics.

The Stacks Project

Open multi-collaboration on foundations of algebraic geometry starting from scratch (category theory/algebra).

Well-organized structure (easy-to-manage dependency graph).

Verified thoroughly for correctness.

The Stacks Project

The Stacks Project now consists of

- 547156 lines of code
- 16738 tags (57 inactive tags)
- 2691 sections
- 99 chapters
- 5712 pages
- 162 slogans

API to query!

- Statements (LaTeX)
- Data for graphs

Tag 01WC

Lemma 28.41.5. *A locally projective morphism is proper.*

code

Proof. Let $f : X \to S$ be locally projective. In order to show that f is proper we may work locally on the base, see Lemma 28.39.3. Hence, by Lemma 28.41.4 above we may assume there exists a closed immersion $X \to \mathbf{P}^n_S$. By Lemmas 28.39.4 and 28.39.6 it suffices to prove that $\mathbf{P}^n_S \to S$ is proper. Since $\mathbf{P}^n_S \to S$ is the base change of $\mathbf{P}^n_{\mathbf{Z}} \to \mathrm{Spec}(\mathbf{Z})$ it suffices to show that $\mathbf{P}^n_{\mathbf{Z}} \to \mathrm{Spec}(\mathbf{Z})$ is proper, see Lemma 28.39.5. By Constructions, Lemma 26.8.8 the scheme $\mathbf{P}^n_{\mathbf{Z}}$ is separated. By Constructions, Lemma 26.8.9 the scheme $\mathbf{P}^n_{\mathbf{Z}}$ is quasi-compact. It is clear that $\mathbf{P}^n_{\mathbf{Z}} \to \mathrm{Spec}(\mathbf{Z})$ is locally of finite type since $\mathbf{P}^n_{\mathbf{Z}}$ is covered by the affine opens $D_+(X_i)$ each of which is the spectrum of the finite type $\mathbf{Z}$-algebra

$$\mathbf{Z}[X_0/X_i, \ldots, X_n/X_i].$$

Finally, we have to show that $\mathbf{P}^n_{\mathbf{Z}} \to \mathrm{Spec}(\mathbf{Z})$ is universally closed. This follows from Constructions, Lemma 26.8.11 and the valuative criterion (see Schemes, Proposition 25.20.6). □

[<< Previous tag] [Next tag >>]

Comments (0) <<<

There are no comments yet for this tag.

Add a comment on tag 01WC >>>

Navigating results

[<< Previous tag] [Next tag >>]

Your location

You're at

- Lemma 41.5 on page 95 of Chapter 28: Morphisms of Schemes
- Lemma 28.41.5 on page 2158 of the book
- lines 9778–9781 of morphisms.tex

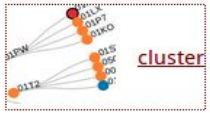
How can you cite this tag? >>>

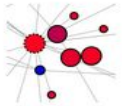
Use:

```
\cite[Tag 01WC]{stacks-project}
```

Extras

- statistics
- history
- dependency graphs:


cluster

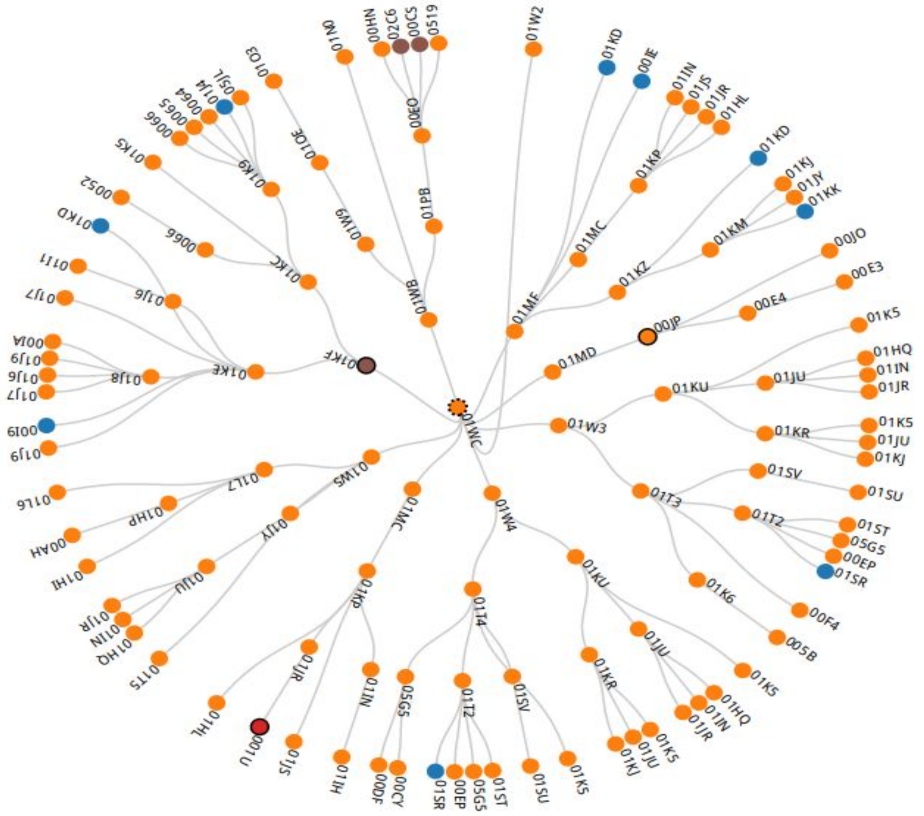

force-directed


collapsible

Tag 01WC points to Lemma 28.41.5
It is contained in Projective morphisms, Chapter 28: Morphisms of Schemes
It has 176 descendant tag(s)

Lemma 28.41.5. *A locally projective morphism is proper.*

Tag 01WC (show tag, view collapsible, view force-directed)



Legend for the type mapping
- Lemma
- Definition
- Section
- Proposition

- this tag has a name
- root

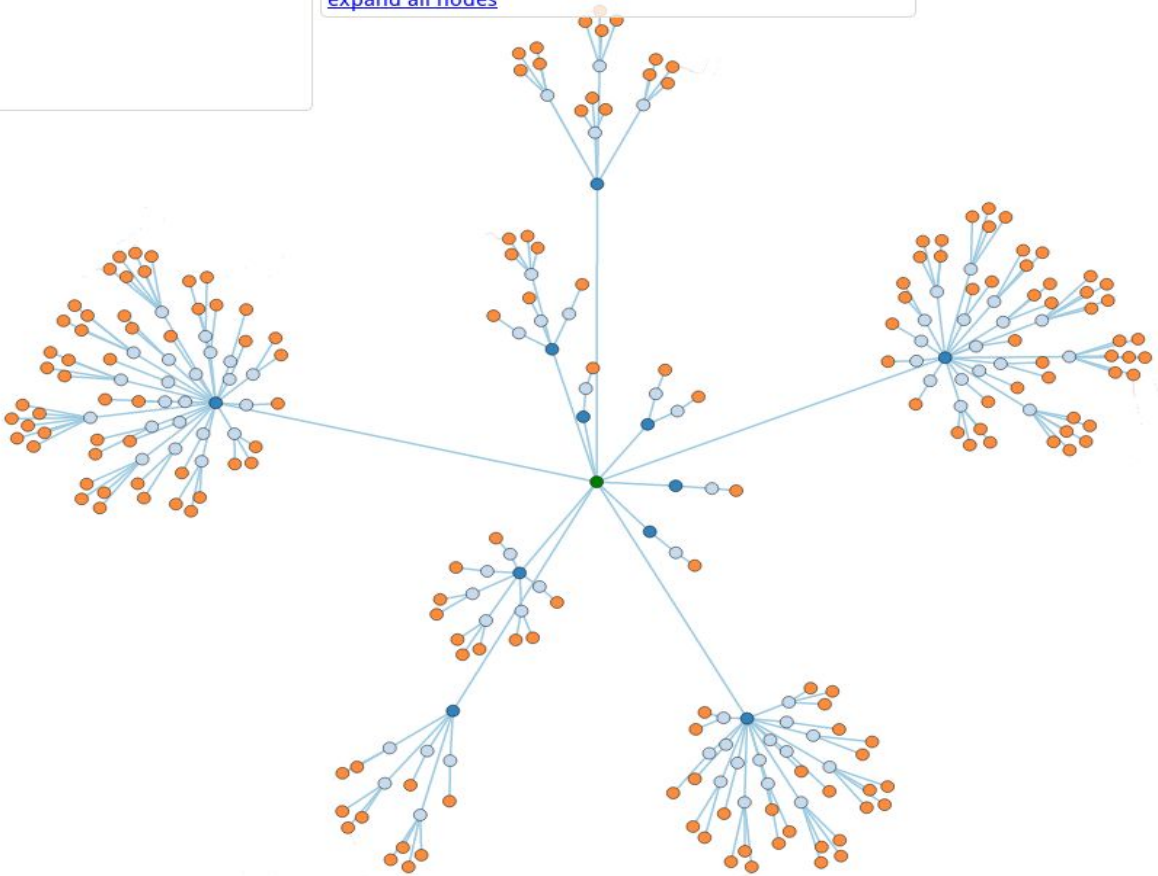
Use the mouse, Luke

Tag 01WC (show tag, view clustered, view force-directed)

Action for tooltip:

(•) preview tag ( ) only tag information ( ) none

expand all nodes



Legend
- Root
- Chapter
- Section
- Tag

DeepAlgebra - an outline

1. Build a dictionary (syntactic parser with Types/variables)
2. Test it on the Stacks Project (build an “ontology” of algebraic geometry)
3. Verify, modify, test it on arXiv (Algebraic Geometry submissions)

Thank you for your attention!