sTEX3 – A LATEX-based Ecosystem for Semantic/Active Mathematical Documents

Dennis Müller & Michael Kohlhase

TUG, 24. July 2022

In mathematics and adjacent disciplines

LaTeX is *the* standard for typesetting documents

- TeX was explicitly conceived for typesetting mathematical formulae
- TeX-like syntax is obiquitous even outside of LaTeX-systems (Wikipedia, Wordpress-Plugins, MathJax,...)
- Scientific Publishing (`arxiv.org`, easychair, Springer,...)

But:

- LaTeX is *only* concerned with the layout of *static* documents (i.e. pure **presentation** of content)
- Barely support for modern dynamic/active document elements (HTML)
- The goal of a document is to transfer the *semantics* of its contents from one brain to other brains.

We can do better!

Formal Mathematics

We know how to represent the precise formal *semantics* of mathematical statements in a machine-actionable manner!

- Basic ideas go back to the early 20th century (Frege, Russell, Hilbert, Gödel,...)
- Allows for automated and interactive theorem proving (Computer-verified mathematics!)

So why don't we just write all math that way?

- Languages reminiscent of programming language, (Completely unrelated to "informal" formulae, complicated syntax, "barely legible" (Lawrence Paulson))
- often counterintuitive for "informal" mathematicians,
- require (almost) *full formalization* of all prerequisite concepts, ($\Rightarrow$ huge "dependency cost")

$\Rightarrow$ difficult to learn and use, require huge investments in terms of expert person hours.

The Pythagorean Theorem in Five Formal Systems

Real math: In any right triangle with sides a, b, c and right angle opposite to c, the equation $a^2 + b^2 = c^2$ holds.

HOL Light:

```
|- !A B C:real^2.
     orthogonal (A - B) (C - B)
     ==> norm(C - A) pow 2 = norm(B - A)
     pow 2 + norm(C - B) pow 2
```

Isabelle:

```
lemma pythagoras:
  fixes a b c :: "'a :: real_inner"
  assumes "orthogonal (b - a) (c - a)"
  shows   "dist b c ^ 2 = dist a b ^ 2 +
    dist a c ^ 2"
```

Coq:

```
Theorem Pythagore :
 forall A B C : PO,
 orthogonal (vec A B) (vec A C) <->
 Rsqr (distance B C) = Rsqr (distance A B)
  + Rsqr (distance A C) :>R.
```

Mizar:

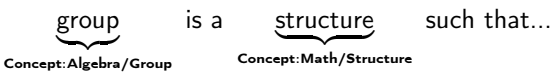
```
theorem :: EUCLID_3:46
  for p1,p2,p3 st p1<>p2 & p3<>p2 &
    (angle(p1,p2,p3)=PI/2 or
      angle(p1,p2,p3)=3/2*PI) holds
    (|.p1-p2.|^2+|.p3-p2.|^2=|.p1-p3.|^2);
```

Metamath: $\vdash ((APB) = 0 \rightarrow ((N'(AGB)) \uparrow 2) = (((N'A) \uparrow 2) + ((N'B) \uparrow 2)))$

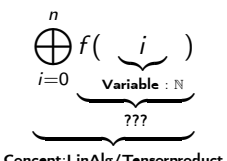
Flexiformal Mathematics

Obervation: The difficulties of formalizing mathematics largely disappear if we switch to *flexiformality*, e.g.

- Semantic annotations for (informal) text fragments:



- Formal expressions with informal/opaque subelements:



While only sacrificing the *strongest* forms of computer-support (i.e. Theorem proving)

A whole suite of computer services remain possible – especially in **active documents**! (disambiguation, quick recaps, guided tours, collaborative library development, structured proofs, reusability,...)

Introducing sTeX

sTeX allows for integrating *semantic annotations* into arbitrary LaTeX documents, covering the full spectrum from informal to fully formal content, and producing *active documents* augmented by semantically informed services.

- The sTeX package allows for declaring semantic macros for semantic markup, organized in a module system. ($\Rightarrow$ Collaborative and communal library development)
- The RusTeX system can convert LaTeX documents to XHTML, preserving both the document layout and the semantic annotations in parallel.
- The MMT system can import the generated XHTML file, extract and interpret the semantic annotations, and host the XHTML as an *active* document with integrated services acting on the semantic annotations.

Active Documents available at `https://mmt.beta.vollki.kwarc.info/:sTeX` (Including 3000+ pages of semantically annotated course notes and slides, libraries with $\geq$ 2250 concepts in Math/CS and (so far) three research papers)

Basic sTeX Concepts

- sTeX allows for declaring symbols with semantic macros and (optional, arbitrarily many) notations attached:
  - `\symdecl{Nat}[name=natural-number]` introduces a new symbol `natural-number` with semantic macro `\Nat` (semantic macros can also take arguments, of course)
  - `\notation{Nat}{\mathbb{N}}` provides it with the notation $\mathbb{N}$
- Symbols are collected in *modules* (`\begin{smodule}{<name>}`), which in turn are collected in archives.
- Modules can be *used* or *imported* via `\usemodule[<archive>]{<module>}`, (Analogous to LaTeX-packages)
  and can bundle closely related symbols, definitions, fundamental theorems, and documentation.

⇒ Modules and Archives can be developed collaboratively and communally!

⇒ *Separation of concerns*: Merely *declaring* new symbols and *using* them subsequently is relatively easy. "Fully formally" annotating them is potentially *hard* (but not required and can be left for others to do)

RusTeX

Many systems allow for converting LaTeX to HTML (10+ actively maintained in 2022)
...none of them were compatible with arbitrary LaTeX packages or attaching arbitrary attributes to arbitrary HTML nodes (...that we tried)

⇒RusTeX: Yet another LaTeX-to-HTML converter, implemented in Rust (sorry)

- *Only* implements primitives of plain TeX, eTeX and pdfTeX
- Processes a user's `latex.ltx` first

⇒ Direct mapping from TeX-whatsits to CSS-classes/nodes (No special treatment of LaTeX macros (sections, lists,...)) (Special treatment for non-primitive macros possible though)

But: No special treatment for sTeX either!

⇒ any converter will do, provided it offers "basic" functionality for attribute annotations! (Uses a backend `cfg`-file)

MMT

MMT: A software system and Scala/Java-API for (flexi-)formal knowledge management services

- Implements services generically and foundation-independent (Library management, parsing, building/compiling, type checking/inference, term simplification, translating, serving/browsing content,...)
- Services dependent on formal foundations (e.g. simplification, type checking) are parametric in *rules* implemented in Scala/Java
- Uses OMDOC/OPENMATH as a backend language (XML)
- Extracts semantic annotations from STEX-generated HTML and translates them into MMT/OMDOC-declarations

⇒ MMT services enabled for STEX-documents

⇒ Can serve the HTML as an *active* document

- An STEX-IDE in the form of a VS Code-Plugin integrates both MMT and RUSTEX directly

Demonstration

There's more to come...

- Change notations and terms in an HTML document based on *reader's* preferences/background (e.g. $\rightarrow$ instead of $\supset$, $\neg$ instead of $\sim$,...)
- *User Models:* Model a student's (or other user's) knowledge state as a probabilistic model, updated on interacting with *knowledge items* (after reading texts, taking small tests,...)

$\Rightarrow$ we can generate guided tours *tailored to a user's prior knowledge or preferences* (Use appropriate examples, a familiar programming language for code snippets, recommend "close"/related topics...)

- *Holy grail:* Integrate sTeX with existing software for mathematics (computer algebra systems, theorem provers, ...) (Have sTeX library serve as a *mediator* between systems: *Math-in-the-Middle*-approach)

Thanks for listening!