System Description: MMTTeX

Connecting Content and Narration-Oriented Document Formats

Florian Rabe

Computer Science, University Erlangen-Nürnberg, Germany
LRI, University Paris-Sud, France

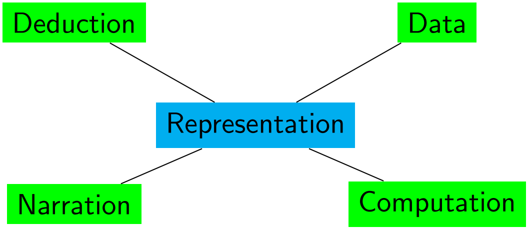
July 2019

Motivation

Tetrapod of Knowledge

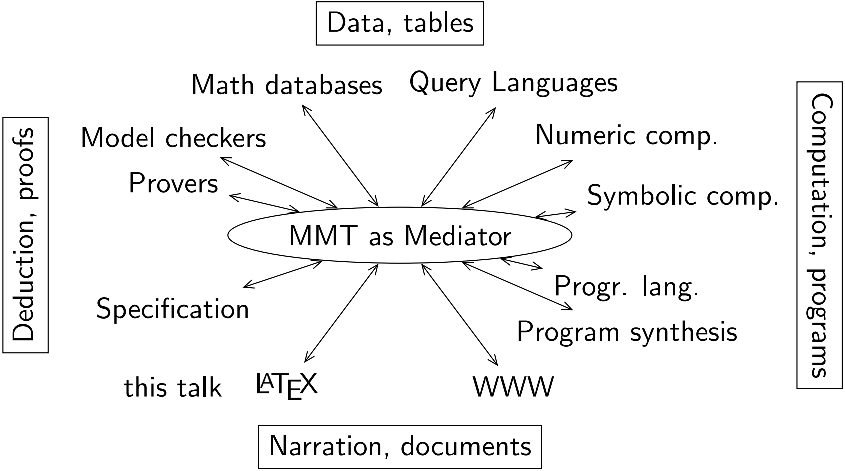
- Narration: informal-but-rigorous math
- Deduction: logic and type systems
- Computation: algorithms
- Data: tables for large sets and functions
- Representation: content dictionaries, ontologies

essential for inter-operability



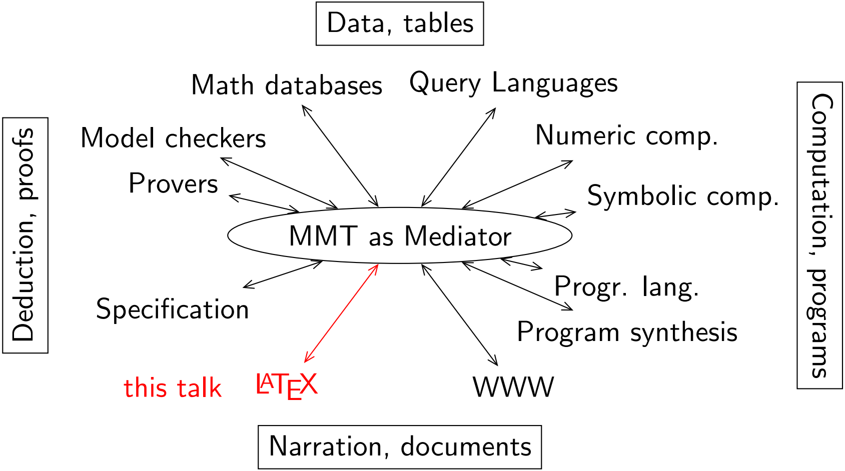
MMT as a System Integration Platform

All system interfaces formalized in MMT
→ semantics-aware tool integration while maintaining existing work flows



MMT as a System Integration Platform

All system interfaces formalized in MMT
→ semantics-aware tool integration while maintaining existing work flows



Design

Ideal System

Requirements

- Authors can mix (at least) MMT and LaTeX in the same file
  - multiple nesting levels
  - top level can be either format
- Control passes between MMT and LaTeX processor
  - sharing the same context
  - communicating context changes
- Produces OMDoc, pdf, HTML, etc.

Problems

- No way to get LaTeX processor to interact dynamically with other systems
- No way to write a new LaTeX processor for the occasion

Realistic Options

Symmetric

- new document format with alternating/nested MMT and LaTeX chunks
- generate `.tex` and `.mmt` files, process separately, merge the outputs

failed 2016 CICM submission, difficult but still interesting

MMT-led

- .mmt file with interspersed LaTeX chunks
- MMT generates `.tex` file

future work

LaTeX-led

- .tex file with interspersed MMT chunks
- LaTeX generates `.mmt` file

this talk

Work flow

BibTeX model:

	Input	Processor	Output
Step 1	`d.tex`	LaTeX	`d.pdf` `d.tex.mmt`
Step 2	`d.tex.mmt`	MMT	`d.tex.omdoc` `d.tex.sty`
Step 3	Run LaTeX again		

2 components:

- `mmttex.sty` package for LaTeX
  - Step 1: writes out MMT chunks to `d.tex.mmt`
  - Step 3: replaces MMT chunks with code from `d.tex.sty`
- `latex-mmt` plugin for MMT
  - Step 2: processes `d.tex.mmt`, generates `d.tex.sty`
  - once at beginning: generates `.sty` files for any other MMT content

Easy to Integrate with Existing Work Flows

One extra LaTeX package

- no conflicts with other packages
- no dependency on LaTeX editor

One extra shell command

- run MMT as black box
- easy to integrate with makefiles, editor shortcuts

Documents re-compilable without MMT

- just include `d.tex.sty` when uploading sources
- running LaTeX still produces `d.tex.mmt` but it has no effect

needed for academic publication

Enables Semantic Formulas inside LaTeX

Semantics processing of `.tex` files

- MMT parsing and type-checking during LaTeX compilation

  semantic errors produce LaTeX errors
- formulas enriched with inferred information

  implicit arguments, omitted types

Semantically enriched formulas in `.pdf`

- tooltips with variable types
- hyperlinks from symbol usage to definition
- whatever else we can get the pdf viewers to support

  e.g., pdf JavaScript exists but barely supported

Example and Demo

3 kinds of MMT content

Kind	defined in	function
Pres.-rel. chunks	LaTeX document	payload
Pres.-irrel. chunks		needed by payload
Backgr. Knowledge	elsewhere	

- Presentation-**relevant** MMT chunks
  - formulas written in MMT syntax, processed by MMT
  - produce semantically enriched formulas in the `.pdf` file

e.g., $2 + x$

- Presentation-**irrelevant** MMT chunks
  - provide context for the pres.-rel. chunks
  - part of `.tex` file
  - no effect on `.pdf` file

e.g., type of x

- Background knowledge
  - available in MMT independent of LaTeX document
  - define formal language(s) used in `tex` file

e.g., definition of $+$

Game Plan

- Background knowledge: typed first-order logic in MMT
- Write a LaTeX document using MMTTeX

  these slides themselves!

  1. define theory of groups
     - informally as usual
     - additional pres.-irrel. chunks for formalization
  2. write formulas about groups in formal MMT syntax

Groups

A group consists of

- a set U,
- an operation $U \to U \to U$, written as infix $*$,
- an element e of U called the unit
- an inverse element function $U \to U$, written as postfix $'$ and with higher precedence than $*$.

We omit the axioms.

Consider group elements a and b.
Then we define the division of a by b as $a{*}b'$.

Division

We extend the theory of groups with a defined operation for division written as a fraction.

Now we can prove $\forall[x]\frac{x}{x}\doteq e$.

Conclusion

Sidenote: Call for Help

How do I make LaTeX forward a Unicode symbol unchanged to the generated `.mmt`?

I have a macro `\toMMT{#1}` that

- appends `#1` to the `.mmt` file
- does not produce any output for the `.pdf` file

But it goes haywire if `#1` contains Unicode characters.

Current workaround:

- avoid Unicode in MMT chunks
- if required by background knowledge, add parsing rules
  e.g., MMT can parse `->` or `\rightarrow` as $\rightarrow$

Prior Attempts

Two predecessors (papers rejected, systems abandoned)

CICM 2013: with M. Iancu, D. Ginev

- also LaTeX-led but with single LaTeX run only
- LaTeX talked to MMT dynamically via HTTP
- main problem: badly chosen story, LaTeX run with `shell-escape` flag

CICM 2016: with M. Iancu, M. Kohlhase, H. Yuan

- symmetric design
- general infrastructure
- arbitrary nesting of MMT and LaTeX
- MMT and LaTeXML output merged into OMDoc
- main problem: complex design hard to implement

Current paper: much narrower focus, much simpler system

Summary

- First step towards integration of LaTeX and MMT

  type-checking while type-setting
- Semantic analysis and enriched presentation of formulas
- Very simple system
- Easy to extend by users
  - new background knowledge using logical frameworks in MMT
  - new LaTeX macros for writing and displaying MMT content
  - variants of MMT plugin for more semantic enrichment
- Future work
  - use MitM as default background knowledge library
  - use MMT also for definitions, proofs, ...
  - combine with MMT-led integration