Active Flexiformal Mathematics (in particular Proofs)

—

Methods, Resources, and Applications

Michael Kohlhase

MCLP; Paris Saclay; 17. 9. 2025

See `https://mathhub.info/?a=mkohlhase%2Ftalks&rp=flexiforms%2Ftalks%2FMCLP25.en.tex` for an active document version.

1 Introduction, (my) Motivation, Conclusion

Instead of an Intro: Disclosure of my Angle Today

- I want to build a universal digital library of mathematical knowledge! (in theory graph form)

Instead of an Intro: Disclosure of my Angle Today

- I want to build a universal digital library of mathematical knowledge! (in theory graph form)
- This is a critical public resource for math systems, education, research,

Instead of an Intro: Disclosure of my Angle Today

- I want to build a universal digital library of mathematical knowledge! (in theory graph form)
- This is a critical public resource for math systems, education, research,
- Without proofs, all of this will be for nothing, (i may get back to this or not)
  - proofs certify the validity of theorems.
  - modularity/reuse via theory morphisms carry proof obligations.
  - ...see all the work in EuroProofNet ...

Instead of an Intro: Disclosure of my Angle Today

- I want to build a universal digital library of mathematical knowledge! (in theory graph form)
- This is a critical public resource for math systems, education, research,
- Without proofs, all of this will be for nothing, (i may get back to this or not)
  - proofs certify the validity of theorems.
  - modularity/reuse via theory morphisms carry proof obligations.
  - ...see all the work in EuroProofNet ...
- **Observation:** Mathematical knowledge and objects are transported by documents!

Instead of an Intro: Disclosure of my Angle Today

- I want to build a universal digital library of mathematical knowledge! (in theory graph form)
- This is a critical public resource for math systems, education, research,
- Without proofs, all of this will be for nothing, (i may get back to this or not)
  - proofs certify the validity of theorems.
  - modularity/reuse via theory morphisms carry proof obligations.
  - . . . see all the work in EuroProofNet . . .
- **Observation:** Mathematical knowledge and objects are transported by documents!
- **Four levels of (electronic) documents:** (numbers are guesswork)

0. printed (for archival purposes) ($\sim$90%)
1. digitized (usually from print) ($\sim$50%)
2. presentational: encoded text interspersed with presentation markup ($\sim$20%)
3. semantic: encoded text with functional markup for the meaning (≤ 0.1%)
4. formal: encoded in some logic (≤ 0.1%)

Transforming down is simple, transforming up needs humans or AI.

Instead of an Intro: Disclosure of my Angle Today

- I want to build a universal digital library of mathematical knowledge! (in theory graph form)
- This is a critical public resource for math systems, education, research,
- Without proofs, all of this will be for nothing, (i may get back to this or not)
  - proofs certify the validity of theorems.
  - modularity/reuse via theory morphisms carry proof obligations.
  - ... see all the work in EuroProofNet ...
- **Observation:** Mathematical knowledge and objects are transported by documents!
- **Four levels of (electronic) documents:** (numbers are guesswork)

  0. printed (for archival purposes) ($\sim$90%)
  1. digitized (usually from print) ($\sim$50%)
  2. presentational: encoded text interspersed with presentation markup ($\sim$20%)
  3. semantic: encoded text with functional markup for the meaning ($\leq$ 0.1%)
  4. formal: encoded in some logic ($\leq$ 0.1%)

  Transforming down is simple, transforming up needs humans or AI.
- **Problem:** We need to deal with mathematical language for math domination.

Instead of an Intro: Disclosure of my Angle Today

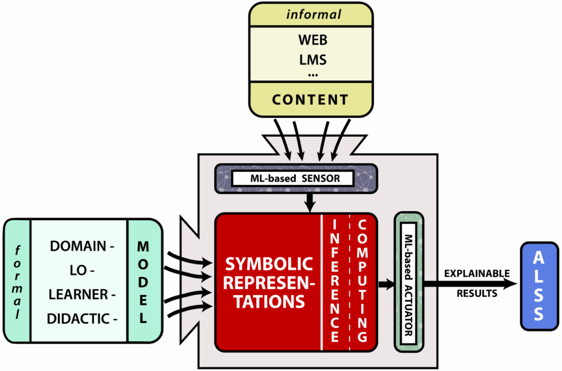
- I want to build a universal digital library of mathematical knowledge! (in theory graph form)
- This is a critical public resource for math systems, education, research,
- Without proofs, all of this will be for nothing, (i may get back to this or not)
  - proofs certify the validity of theorems.
  - modularity/reuse via theory morphisms carry proof obligations.
  - ...see all the work in EuroProofNet ...
- **Observation:** Mathematical knowledge and objects are transported by documents!
- **Four levels of (electronic) documents:** (numbers are guesswork)
  0. printed (for archival purposes) ($\sim$90%)
  1. digitized (usually from print) ($\sim$50%)
  2. presentational: encoded text interspersed with presentation markup ($\sim$20%)
  3. semantic: encoded text with functional markup for the meaning ($\leq$ 0.1%)
  4. formal: encoded in some logic ($\leq$ 0.1%)

  Transforming down is simple, transforming up needs humans or AI.
- **Problem:** We need to deal with mathematical language for math domination.
- **Observation:** For many math support tasks, textual math is enough. (the semantic level)

  But controlled natural language (CNL) is a non-starter! (cannot re-write all math)
- **My current case study:** Univ. Education supported by Symbolic AI . (Adaptive Learning Assistant)

Full Disclosure: My Intuitions about the AI Spectrum?

- Thanks to Stefan Schulz for raising the AI Spectrum question.
- **Definition 1.1.** An AI system is called a symbolic core system, if it uses symbolic representations, symbolic computation, and inference at the core to produce results with explanations. It may use other forms of AI technology to perform sensory and actuatory tasks at the periphery of the system.



- We should invest in symbolic core systems rather than "just ask an LLM" (also cf. EU AI Act)

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\rightsquigarrow$ symbolic core)

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALEA)
  - Automation of flexiformalization is possible/desirable ($\leadsto$ symbolic core)
- **Flexiformalization:**
  - Relax on "verification", gain machine-actionable artifacts.
  - Informal/formal continuum allows incremental formalization.
  - We can keep modularization and proofs for automation. (maybe opaque)
  - Flexiformal representation formats like sTEX mix formal/informal parts.

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALEA)
  - Automation of flexiformalization is possible/desirable ($\leadsto$ symbolic core)
- **Flexiformalization:**
- **Applications:** Incremental formalization/informalization, active documents, education,. . .

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\rightsquigarrow$ symbolic core)
- **Flexiformalization:**
- **Applications:** Incremental formalization/informalization, active documents, education,...
- **Flexiformal Libraries/Workflows:**
  - Focus on building (small) flexiformal artifacts ($\widehat{=}$ elaboration?)
  - TEX parsing, macro expansion takes 95% of the build time. (pdflatex/rusTeX)
  - Separate compilation and document contextualization as a solution?

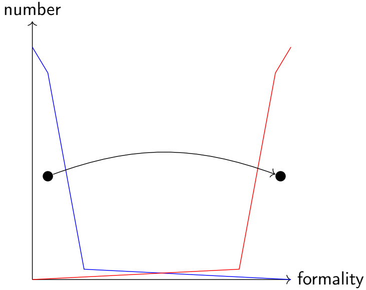
Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\rightsquigarrow$ symbolic core)
- **Flexiformalization:**
- **Applications:** Incremental formalization/informalization, active documents, education,. . .
- **Flexiformal Libraries/Workflows:**
- **Ongoing/Future Work:**
  - Harvest a lexical resource MathLex: `https://github.com/OpenMath/mathlex`
  - Use the theory graph structure for re-usability
  - FAIR (Findable, Accessible, Reusable, Interoperable) Math
  - A flexiformal domain model for undergraduate Math/CS ($\widehat{=}$ MathLib^{-})
  - Semantic services (learning interventions) for ALeA (`https://alea.education`)

2 Flexiformality

Migration by Stepwise Formalization

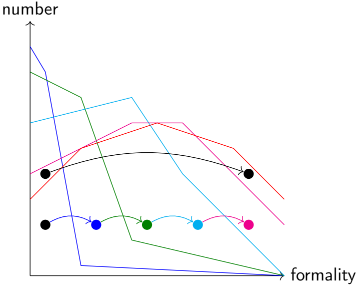
- Full Formalization is hard (we have to commit, make explicit)
- Let's look at documents and document collections.



Currently almost all documents are informal, we would like all formal $\leadsto$ big jump!

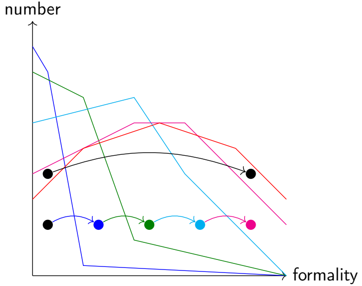
Migration by Stepwise Formalization

- Full Formalization is hard (we have to commit, make explicit)
- Let's look at documents and document collections. Currently almost all documents are informal, we would like all formal $\leadsto$ big jump!
- **Idea:** Allow partial/stepwise formalization



Migration by Stepwise Formalization

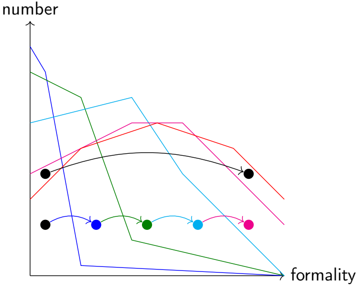
- Full Formalization is hard (we have to commit, make explicit)
- Let's look at documents and document collections. Currently almost all documents are informal, we would like all formal $\leadsto$ big jump!
- **Idea:** Allow partial/stepwise formalization



- **Prerequisite:** A format that allows flexible formalization (which aspects?, how deep?)

Migration by Stepwise Formalization

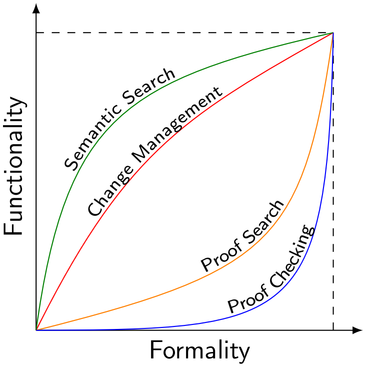
- Full Formalization is hard (we have to commit, make explicit)
- Let's look at documents and document collections. Currently almost all documents are informal, we would like all formal $\leadsto$ big jump!
- **Idea:** Allow partial/stepwise formalization



- **Prerequisite:** A format that allows flexible formalization (which aspects?, how deep?)
- **Opportunity:** Formalization as a continuous process in a controlled environment.

Functionality of Flexiformal Services

- **Generally:** Flexiformal services deliver according to formality level (GIGO: Garbage in $\leadsto$ Garbage out!)
- **But:** Services have differing functionality profiles.
  - Math Search works well on informal documents
  - Change management only needs dependency information
  - Proof search needs theorem formalized in logic
  - Proof checking needs formal proof too



The Flexiformalist Program (Details in [Koh13])

- The development of a regime of partially formalizing
  - mathematical knowledge into a modular ontology of mathematical theories (content commons), and
  - mathematical documents by semantic annotations and links into the content commons (semantic documents),
- The establishment of a software infrastructure with
  - a distributed network of archives that manage the content commons and collections of semantic documents,
  - semantic web services that perform tasks to support current and future mathematic practices
  - active document players that present semantic documents to readers and give access to respective
- The re-development of comprehensive part of mathematical knowledge and the mathematical documents that carries it into a flexiformal digital library of mathematics.

Stephen Watt's understanding of Flexiformality

A person who is flexiformal:

- flexible (contortionist)
- formal (tuxedo)

3 Flexiformal Theory Graphs and Proofs

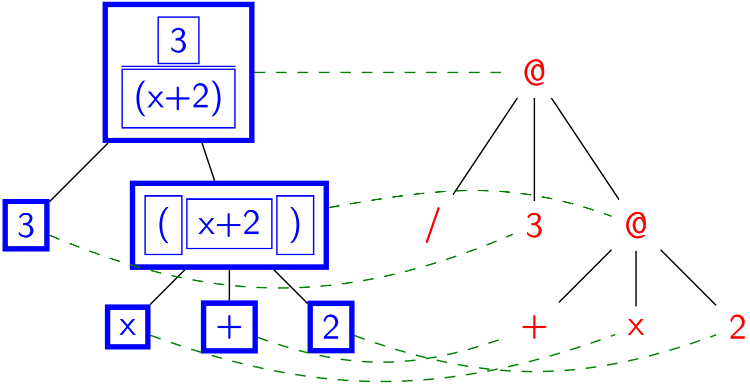
How to model Flexiformal Mathematics

- **I hope to have convinced you:** that Math is informal:
  - foundations unspecified (what a relief)
  - natural language & presentation formulae (humans can disambiguate)
  - context references (but math is better than the pack)
- **Problem:** How do we deal with that in our “formal” systems?
- **Proposed Answer:** learn from *OpenMath*/MathML
  - referential theory of meaning (by pointing to symbol definitions)
  - allow opaque content (presentation/natural language)
  - parallel markup (mix formal/informal recursively at any level)
  - pluralism at all levels (object/logic/foundation/metalogic)
  - underspecification of symbol meaning

  extend to statement/paragraph and theory/discourse levels (OMDoc)

Parallel Markup e.g. in MathML I

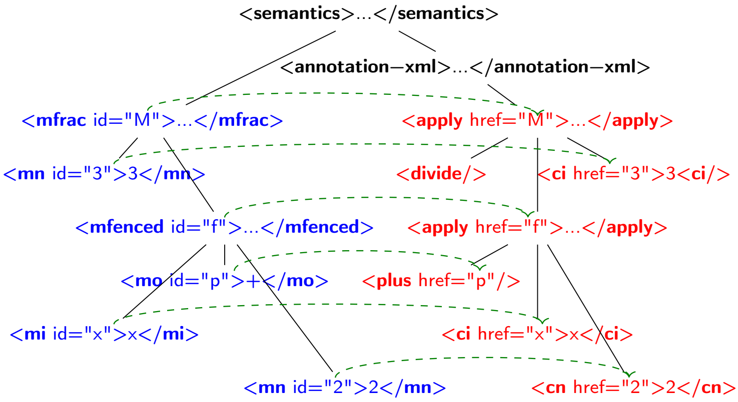
- **Idea:** Combine the presentation and content markup and cross-reference



- use e.g. for semantic copy and paste. (click o3n presentation, follow link and copy content)

Parallel Markup e.g. in MathML II

- **Concrete Realization in MathML:** semantics element with presentation as first child and content in annotation–xml child



Parallel Markup at the Discourse Level

- **Observation:** Parallel markup supports two workflows:
  - *formalization*: the annotation of presentation formulae with (multiple) formalizations and
  - *presentation*: the annotation of content formulae with (multiple) presentations.

Parallel Markup at the Discourse Level

- **Observation:** Parallel markup supports two workflows:
  - *formalization*: the annotation of presentation formulae with (multiple) formalizations and
  - *presentation*: the annotation of content formulae with (multiple) presentations.
- **Idea:** At the discourse level, the duality between presentation and content manifests itself as that between narration and declarations.
  - **in OMDoc:** Statements are logical paragraphs with markup for special relations
  - **in Mmt:** Statements are declarations of the form $c[:\tau][=\delta]$ (via Curry-Howard iso)

Parallel Markup at the Discourse Level

- **Observation:** Parallel markup supports two workflows:
  - *formalization*: the annotation of presentation formulae with (multiple) formalizations and
  - *presentation*: the annotation of content formulae with (multiple) presentations.
- **Idea:** At the discourse level, the duality between presentation and content manifests itself as that between narration and declarations.
  - **in OMDoc:** Statements are logical paragraphs with markup for special relations
  - **in Mmt:** Statements are declarations of the form $c[:\tau][=\delta]$ (via Curry-Howard iso)
- **Example 3.1 (Parallel Markup for a Definition).**

Definition 08.15
The **exponential function** $e : \mathbb{R} \to \mathbb{R}$

is the solution of $f' = f$ with $f(0) = 1$.

exp	:	$\mathbb{R} \to \mathbb{R}$
	$=$	$\tau f.f' = f \wedge f(0) = 1$

Parallel Markup at the Discourse Level

- **Observation:** Parallel markup supports two workflows:
  - *formalization*: the annotation of presentation formulae with (multiple) formalizations and
  - *presentation*: the annotation of content formulae with (multiple) presentations.
- **Idea:** At the discourse level, the duality between presentation and content manifests itself as that between narration and declarations.
  - **in OMDoc:** Statements are logical paragraphs with markup for special relations
  - **in Mmt:** Statements are declarations of the form $c[:\tau][=\delta]$ (via Curry-Howard iso)
- **Example 3.1 (Parallel Markup for a Definition).**

Definition 08.15
The **exponential function** $e : \mathbb{R} \to \mathbb{R}$
is the solution of $f' = f$ with $f(0) = 1$.

$$\begin{array}{lll} \exp & : & \mathbb{R} \to \mathbb{R} \\ & = & \tau f.f' = f \wedge f(0) = 1 \end{array}$$

- **Implementation Idea:** Mix formal and informal in a single format, e.g. sTeX
  - Annotate formal classes/relations in underlying LaTeX text (OMDoc ontology)
  - Semantic macros with mutable notations in formulae (MathML ontology)

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*
- *Proof sketch:* Euclid proved this ca. 300 BC, so we leave it as an exercise

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*
- *Proof sketch:* Euclid proved this ca. 300 BC, so we leave it as an exercise
- *Proof sketch:* Suppose $p_1, p_2, \ldots, p_k$ are all the primes. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P. But every p_i divides $P - 1$ so p cannot be any of them. Therefore p is a new prime. Contradiction, so there are infinitely many primes.

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*
- *Proof sketch:* Euclid proved this ca. 300 BC, so we leave it as an exercise
- *Proof sketch:* Suppose $p_1, p_2, \ldots, p_k$ are all the primes. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P. But every p_i divides $P - 1$ so p cannot be any of them. Therefore p is a new prime. Contradiction, so there are infinitely many primes.
- *Proof:*
  1. Suppose $p_1, p_2, \ldots, p_k$ are all the primes.
  2. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P.
  3. But every p_i divides $P - 1$ so p cannot be any of them.
  4. Therefore p is a new prime.
  5. Contradiction, so there are infinitely many primes.

$\square$

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*
- *Proof sketch:* Euclid proved this ca. 300 BC, so we leave it as an exercise
- *Proof sketch:* Suppose $p_1, p_2, \ldots, p_k$ are all the primes. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P. But every p_i divides $P - 1$ so p cannot be any of them. Therefore p is a new prime. Contradiction, so there are infinitely many primes.
- *Proof:*
  1. Suppose $p_1, p_2, \ldots, p_k$ are all the primes.
  2. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P.
  3. But every p_i divides $P - 1$ so p cannot be any of them.
  4. Therefore p is a new prime.
  5. Contradiction, so there are infinitely many primes.

  □
- **Flexiformal Proof term:** $\bot E(\lambda X :\vdash (p_1, p_2, \ldots, p_k \text{ are } \ldots). \ldots \bot))$ (formal justifications, opaque types)

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*
- *Proof sketch:* Euclid proved this ca. 300 BC, so we leave it as an exercise
- *Proof sketch:* Suppose $p_1, p_2, \ldots, p_k$ are all the primes. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P. But every p_i divides $P - 1$ so p cannot be any of them. Therefore p is a new prime. Contradiction, so there are infinitely many primes.
- *Proof:*
  1. Suppose $p_1, p_2, \ldots, p_k$ are all the primes.
  2. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P.
  3. But every p_i divides $P - 1$ so p cannot be any of them.
  4. Therefore p is a new prime.
  5. Contradiction, so there are infinitely many primes.

  $\square$
- **Flexiformal Proof term:** $\bot E(\lambda X :\vdash (p_1, p_2, \ldots, p_k \text{ are } \ldots) \ldots . \bot))$ (formal justifications, opaque types)
- Fully formal proof term in LF: $\bot E(\lambda X :\vdash (\ldots) \ldots . \bot))$

Flexiformal Proofs by Opaqueness and Parallel Markup

- **Theorem 3.2.** *There are infinitely many primes.*
- *Proof sketch:* Euclid proved this ca. 300 BC, so we leave it as an exercise
- *Proof sketch:* Suppose $p_1, p_2, \ldots, p_k$ are all the primes. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P. But every p_i divides $P - 1$ so p cannot be any of them. Therefore p is a new prime. Contradiction, so there are infinitely many primes.
- *Proof:*
  1. Suppose $p_1, p_2, \ldots, p_k$ are all the primes.
  2. Then, let $P = \Pi_{i=1}^{k} p_i + 1$ and p a prime dividing P.
  3. But every p_i divides $P - 1$ so p cannot be any of them.
  4. Therefore p is a new prime.
  5. Contradiction, so there are infinitely many primes.

  $\square$

- **Flexiformal Proof term:** $\bot E(\lambda X :\vdash (p_1, p_2, \ldots, p_k \text{ are } \ldots).\ldots\bot))$ (formal justifications, opaque types)
- Fully formal proof term in LF: $\bot E(\lambda X :\vdash (\ldots).\ldots\bot))$
- Cf. Claudio et al's paper on mapping OMDoc proofs into $\bar{\lambda}\mu\widetilde{\mu}$-calculus [ASC06]

4 The “STEM Education” Application in a Nutshell

Mechanize what Good Teachers Do (Four Models in ALeA)

- **Idea:** Mechanize what good teachers do! (relieve them from routine tasks)

Mechanize what Good Teachers Do (Four Models in ALeA)

- **Idea:** Mechanize what good teachers do! (relieve them from routine tasks)
- **Claim:** Good teachers use four fine-grained models: one each of
  1. the structure of the underlying domain knowledge (reviewed in course preparation)
  2. how to formulate/exhibit/test that knowledge (tied up in slides/problem collections)
  3. how these formulations play together didactically (teaching process intuitions)
  4. student preferences/competencies (teachers collect/maintain that implicitly)

Mechanize what Good Teachers Do (Four Models in ALeA)

- **Idea:** Mechanize what good teachers do! (relieve them from routine tasks)
- **Claim:** Good teachers use four fine-grained models: one each of
  1. the structure of the underlying domain knowledge (reviewed in course preparation)
  2. how to formulate/exhibit/test that knowledge (tied up in slides/problem collections)
  3. how these formulations play together didactically (teaching process intuitions)
  4. student preferences/competencies (teachers collect/maintain that implicitly)

  For explanations/courses teachers...

  from 4. decide what the student needs to be told (and what to leave out)

  from 1. decide how to structure the explanation/course (at what level)

  from 3. select motivations, introductions, transitions, examples, proofs, ...

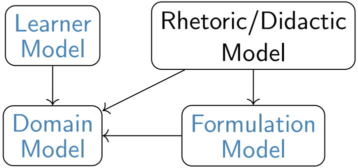
  from 2. assemble structured, coherent "document" from existing resources.

Mechanize what Good Teachers Do (Four Models in ALEA)

- **Idea:** Mechanize what good teachers do! (relieve them from routine tasks)
- **Claim:** Good teachers use four fine-grained models: one each of
  1. the structure of the underlying domain knowledge (reviewed in course preparation)
  2. how to formulate/exhibit/test that knowledge (tied up in slides/problem collections)
  3. how these formulations play together didactically (teaching process intuitions)
  4. student preferences/competencies (teachers collect/maintain that implicitly)
- **Intuition:** Given such models we can mechanize explanations/courses (+ more)
- **Necessary Investment:** To obtain these, we must (1.-3. offline, 4. online)
  1. formalize domain knowledge in a fine-grained knowledge graph, (concepts with IDs)
  2. annotate learning objects (definitions, remarks, problems,...) with concept IDs,
  3. classify and interrelate learning object via their IDs and with concept IDs,
  4. record student competencies can be relative to concept IDs.

Mechanize what Good Teachers Do (Four Models in ALEA)

- **Idea:** Mechanize what good teachers do! (relieve them from routine tasks)
- **Claim:** Good teachers use four fine-grained models: one each of
  1. the structure of the underlying domain knowledge (reviewed in course preparation)
  2. how to formulate/exhibit/test that knowledge (tied up in slides/problem collections)
  3. how these formulations play together didactically (teaching process intuitions)
  4. student preferences/competencies (teachers collect/maintain that implicitly)
- **Intuition:** Given such models we can mechanize explanations/courses (+ more)
- **Necessary Investment:** To obtain these, we must (1.-3. offline, 4. online)
  1. formalize domain knowledge in a fine-grained knowledge graph, (concepts with IDs)
  2. annotate learning objects (definitions, remarks, problems,. . .) with concept IDs,
  3. classify and interrelate learning object via their IDs and with concept IDs,
  4. record student competencies can be relative to concept IDs.
- **Conceptual Architecture:** (arrows $\widehat{=}$ concept ID references)
  1. Domain Model $\widehat{=}$ formal/MMT theory graph
  2. Formulation model $\widehat{=}$ flexiformal/sTeX text fragments
  3. R/D Model $\widehat{=}$ RDF Metadata
  4. Learner Model : Learners $\times$ Concepts $\to$ Competencies



Introducing sTeX- a LaTeX-based Flexiformal Language

- 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 theory graph. ($\leadsto$ 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.
  - Workflow bundled into / hidden by a VSCode plugin

Introducing sTeX- a LaTeX-based Flexiformal Language

- 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 theory graph. ($\leadsto$ 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.
  - Workflow bundled into / hidden by a VSCode plugin
- Active Documents available at `https://mathhub.info/dashboard/mathhub` (Including 3000+ pages of semantically annotated course notes and slides, libraries with $\geq$ 2500 concepts in Math/CS and (so far) three research papers)
- Course portal based on sTeX documents: `http://alea.education`

Example: sTeX Modules from the Domain Model

```
\documentclass{stex}
\begin{document}
\begin{smodule}{sets}\symdef{member}[args=ai]{#1\maincomp\in #2}\end{smodule}

\begin{smodule}{magma}
  \importmodule{sets}
  \symdef{sset}{\comp{\mathcal{S}}} % the base set
  \symdef{sop}[args=2]{(#1\maincomp\circ #2)} % operation
  \symdecl*{magma}
  \begin{sdefinition}[id=magma.def]
    A structure $\mathstruct{\sset,\sop!}$ is called a \definame{magma}, if $\sset$ is closed
    under $\sop!$, i.e. if $\member{\sop{a}b}\sset$ for all $\member{a,b}\sset$.
  \end{sdefinition}
\end{smodule}

\begin{smodule}{semigroup}
  \importmodule{magma}
  \symdecl*{semigroup}
  \begin{sdefinition}[id=semigroup.def]
    A \sn{magma} $\mathstruct{\sset,\sop!}$ is called a \definame{semigroup}, if
    $\sop!$ is associative on $\sset$, i.e. if $\sop{a}{\sop{b}{c}}=\sop{\sop{a}{b}}{c}$
    for all $\member{a,b,c}\sset$.
  \end{sdefinition}
\end{smodule}
```

Example; Multilinguality in sTeX

- **Note:** If we translate, we do not have to duplicate formal parts and change formulae.

Example; Multilinguality in sTeX

- **Note:** If we translate, we do not have to duplicate formal parts and change formulae.
- **Example 4.1 (The sTeX Modules above in German).**

```
% no module sets <-- no vocabulary

\begin{smodule}[sig=en]{magma}
  % no \import*, no \symdef, no \symdecl*; see the english module
  \begin{sdefinition}[id=magma.def]
    Ist $\sset$ is abgeschlossen unter $\sop!$, d.h. ist $\member{\sop{a}b}\sset$ fuer
    alle $\member{a,b}\sset$, so nennen wir eine Struktur $\mathstruct{\sset,\sop!}$ ein
    \Definame{magma}.
  \end{sdefinition}
\end{smodule}

\begin{smodule}[sig=en]{semigroup}
  \begin{sdefinition}[id=semigroup.def]
    Ein \Sn{magma} $\mathstruct{\sset,\sop!}$ heisst \definiendum{semigroup}{Halbgruppe},
    wenn $\sop!$ assoziativ auf $\sset$ ist, d.h. wenn
    $\sop{a}{\sop{b}{c}}=\sop{\sop{a}{b}}{c}$ fuer alle $\member{a,b,c}\sset$.
  \end{sdefinition}
\end{smodule}
```

- **Note:** Definienda are pairs of system name (a URI internally) and a verbalization. (here in German)

Example; Multilinguality in sTEX

- **Note:** If we translate, we do not have to duplicate formal parts and change formulae.
- **Example 4.1 (The sTEX Modules above in German).**

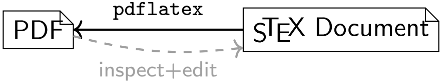
```
% no module sets <-- no vocabulary

\begin{smodule}[sig=en]{magma}
  % no \import*, no \symdef, no \symdecl*; see the english module
  \begin{sdefinition}[id=magma.def]
    Ist $\sset$ is abgeschlossen unter $\sop!$, d.h. ist $\member{\sop{a}b}\sset$ fuer
    alle $\member{a,b}\sset$, so nennen wir eine Struktur $\mathstruct{\sset,\sop!}$ ein
    \Definame{magma}.
  \end{sdefinition}
\end{smodule}

\begin{smodule}[sig=en]{semigroup}
  \begin{sdefinition}[id=semigroup.def]
    Ein \Sn{magma} $\mathstruct{\sset,\sop!}$ heisst \definiendum{semigroup}{Halbgruppe},
    wenn $\sop!$ assoziativ auf $\sset$ ist, d.h. wenn
    $\sop{a}{\sop{b}{c}}=\sop{\sop{a}{b}}{c}$ fuer alle $\member{a,b,c}\sset$.
  \end{sdefinition}
\end{smodule}
```

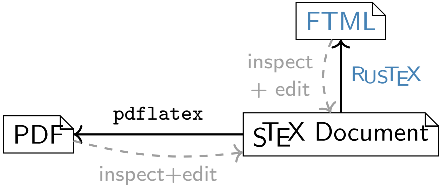
- **Note:** Definienda are pairs of system name (a URI internally) and a verbalization. (here in German)
- **State:** ∼70% of modules in German, ∼6% in Chinese, some Turkish, Romanian, . . .

STEX Workflow (for flexiformal course materials)



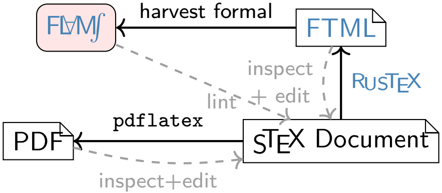
Step 1. Develop STEX course materials classically

sTEX Workflow (for flexiformal course materials)



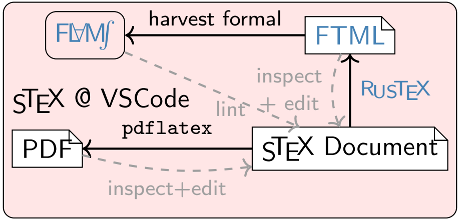
Step 2. Convert to FTML via RusTEX

sTeX Workflow (for flexiformal course materials)



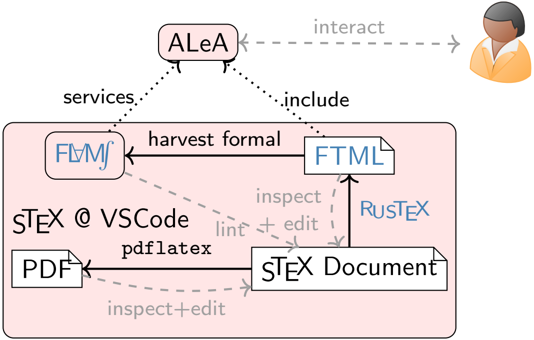
Step 3. Harvest semantics into the FLAMS system

sTEX Workflow (for flexiformal course materials)



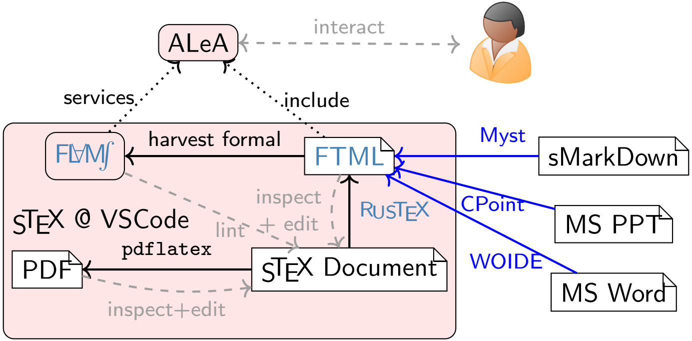
BTW: All of this in an IDE

sTEX Workflow (for flexiformal course materials)



Step 4. Import into ALeA (Math UI for Adaptive Learning)

sTEX Workflow (for flexiformal course materials)



New: More Sources of FTML for authors without LATEX

5 Towards a Common Lexical Resource for Mathematics

Towards a Common Lexical Resource for Mathematics

- **Problem:** The vocabulary of Math is humongous, balkanized, and heavily overloaded.

Towards a Common Lexical Resource for Mathematics

- **Problem:** The vocabulary of Math is humongous, balkanized, and heavily overloaded.
  $\leadsto$ if every journal paper only defines one new concept, then we have over 4M words/terms.

Towards a Common Lexical Resource for Mathematics

- **Problem:** The vocabulary of Math is humongous, balkanized, and heavily overloaded.
  $\leftrightsquigarrow$ if every journal paper only defines one new concept, then we have over 4M words/terms.
  $\leftrightsquigarrow$ The MSC 2020 lists $\sim$ 7000 separate sub-areas in Math, most of which know nothing about each other.

Towards a Common Lexical Resource for Mathematics

- **Problem:** The vocabulary of Math is humongous, balkanized, and heavily overloaded.
  $\leftrightsquigarrow$ if every journal paper only defines one new concept, then we have over 4M words/terms.
  $\leftrightsquigarrow$ The MSC 2020 lists $\sim$ 7000 separate sub-areas in Math, most of which know nothing about each other.
  $\leadsto$ therefore there is heavy re-use of (the $\leq$ 0.5M) words of e.g. English.
- **More $\leadsto$ more Problems:** The vocabulary of Math is
  1. multilingual: we can write math in any language
  2. many mathematical terms also have notations for presentation formulae
  3. synonyms, homonyms, homographs, "house styles" abound
- **Idea:** Collect a lexical/semantic resource together with the mathematical knowledge/documents. (at least for canonical (KG-B.Sc) math)
- **Plan:** (so far Aarne Ranta, Frederik Schaefer, and me)
  1. Fix a simple representation format that everyone can generate.
  2. Export information from all sources of lexical information.
  3. Publish early, publish often!

Lexical Aspects of Mathematical Language – Verbs

- Verbs are relatively rare in Mathematics:
  - "*converges*" ("*pointwise*"), "*divides*", "*intersects*" (verbalizations; do not really count)
  - "*. . . A is a B*", "*Let . . . be a . . .*", (foundational)
  - "*We have*", "*consider*"/"*assume*" (foundational, but for argumentation/proof)

Lexical Aspects of Mathematical Language – Adjectives

- Adjectives come in three (semantic) categories:
  - "*odd*", "*prime*", etc. (intersective $\widehat{=}$ An odd integer is in *odd* $\cap$ *integer*)
  - "*simple group*", "*discrete topology*" (subsective $\widehat{=}$ a simple group is still a group (but not intersective))
  - "*simple group*" vs. "*simple cycle*" (homonyms $\widehat{=}$ different adjectives?)
  - "*partial function*", "*contravariant functor*" (no longer subsective $\rightsquigarrow$ general set transformer)
- **My Suspiciion:** Mathematicians seem to hate non-subsective adjectives
  $\rightsquigarrow$ make partial function the general case and function the specialization.

Lexical Aspects of Mathematical Language – Adjectives

- Simple nouns – i.e. without inner structure – are relatively boring
- Named entities, e.g. "*Kripke-structure*", "*Fundamental Theorem of Algebra*",...

Lexical Aspects of Mathematical Language – Adjectives

- Simple nouns – i.e. without inner structure – are relatively boring
- Named entities, e.g. "*Kripke-structure*", "*Fundamental Theorem of Algebra*",...
- Functional nouns (Noun Constructors) (much more interesting)
  - "*the natural logarithm of x*"
  - "*the sum of a(, b,) and c*" vs. "*a plus b plus c*"
  - "*the quotient space of $\mathbb{Z}$ over $n\mathbb{Z}$*" vs. "*$\mathbb{Z}$ mod $n\mathbb{Z}$*"
  - "*the general linear group of order n over (the ring) R*".
  - "*the line between A and B*".
  - "*the integral over $f(x)$ from A to B wrt. x*".
  - "*the n-dimensional identity matrix over $\mathbb{Z}$*".

Lexical Aspects of Mathematical Language – Adjectives

- Simple nouns – i.e. without inner structure – are relatively boring
- Named entities, e.g. "*Kripke-structure*", "*Fundamental Theorem of Algebra*",...
- Functional nouns (Noun Constructors) (much more interesting)
  - "*the natural logarithm of x*"
  - "*the sum of a(, b,) and c*" vs. "*a plus b plus c*"
  - "*the quotient space of $\mathbb{Z}$ over $n\mathbb{Z}$*" vs. "*$\mathbb{Z}$ mod $n\mathbb{Z}$*"
  - "*the general linear group of order n over (the ring) R*".
  - "*the line between A and B*".
  - "*the integral over $f(x)$ from A to B wrt. x*".
  - "*the n-dimensional identity matrix over $\mathbb{Z}$*".
- Nouns for Algebraic Structures, e.g. "*group*", "*metric space*", "*vector space*",
  - required/optional arguments in a tuple structure
  - access by accessor concept/method
  - structure extension, instantiation, and interpretation as central operations

Lexical Aspects of Mathematical Language – Predicates/Relations

- Predicates/Relations
  - "*A is an n-ary function*", "*A σ-trivializes B*"
- But Philippe de Groote's talk gave much more information on these phenomena.

Towards a Common Lexical Resource for Mathematics

- **Problem:** The vocabulary of Math is humongous, balkanized, and heavily overloaded.

Towards a Common Lexical Resource for Mathematics

- **Problem:** The vocabulary of Math is humongous, balkanized, and heavily overloaded.
- **Concrete Idea:** for realizing a collectively curated lexical resource
  - use the GF Resource Grammar Library (RGL) infrastructure for grammatical aspects.
  - use the notations/argument specifiers/types from the OMDoc ontology for the semantics.
  - encode in a standard framework like JSON.
- **Concrete Plan:** Generate an initial resource from the sTeX corpus, align/complement with Lean MathLib/WikiData.

Towards a Common Lexical Resource for Mathematics

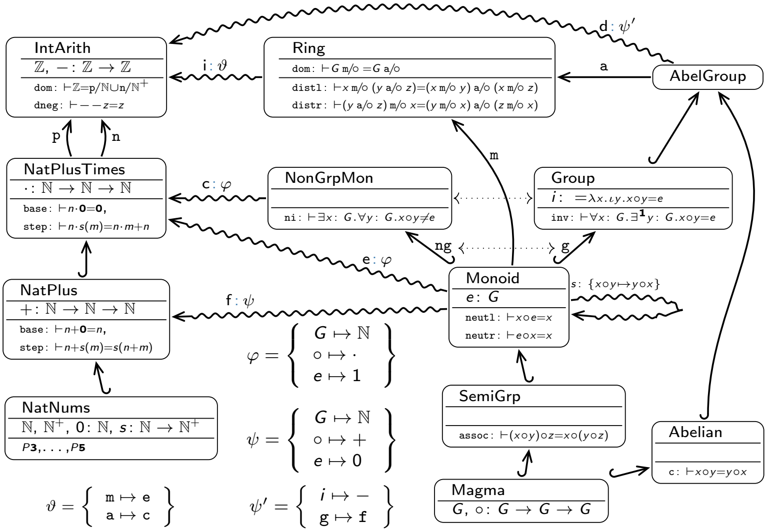
- **Concrete Example:** An entry for the term "*divides*":

```
example_0 = {
    'id': '0',
    'name': 'divides',
    'status': 'experimental',
    'latex': '#1 | #2',
    'stex_sig': 'ii',
    'stex_macro': 'divisor',
    'gf_cat': 'Relverb',
    'dk_type': 'Elem Int -> Elem Int -> Prop',
    'dk_def': 'm => n => exists Int (k => Eq n (times k m))',
    'gf_fun': 'divide_Relverb',
    'gf_examples': {
        'abstract': 'RelverbProp divide_Relverb (TermExp (TNumber 7)) (TermExp (TNumber 91))',
        'Eng': '$7$ divides $91$',
        'Fre': '$7$ divise $91$',
        'Ger': '$7$ teilt $91$',
        'Swe': '$7$ delar $91$'
        },
    'raw_examples': {
        'Fin': '$7$ jakaa $91$:n'
        },
    'alignments': {
        'wikidata': 'https://www.wikidata.org/wiki/Q50708',
        'stex': 'https://mathhub.info?a=smglom/arithmetics&p=mod&m=divisor&s=divisor',
        'lean': 'https://leanprover-community.github.io/mathlib4_docs/Mathlib/GroupTheory/Divisible.html#DivisibleBy'
        }
    }
```

6 Using Theory Graphs Profitably in Education

Modular Representation of Math (MMT Example)

- **Example 6.1 (Elementary Algebra and Arithmetics).**



$$\varphi = \left\{ \begin{array}{l} G \mapsto \mathbb{N} \\ \circ \mapsto \cdot \\ e \mapsto 1 \end{array} \right\}$$

$$\psi = \left\{ \begin{array}{l} G \mapsto \mathbb{N} \\ \circ \mapsto + \\ e \mapsto 0 \end{array} \right\}$$

$$\vartheta = \left\{ \begin{array}{l} \mathtt{m} \mapsto \mathtt{e} \\ \mathtt{a} \mapsto \mathtt{c} \end{array} \right\} \qquad \psi' = \left\{ \begin{array}{l} i \mapsto - \\ \mathtt{g} \mapsto \mathtt{f} \end{array} \right\}$$

Background: Redesigning OMDoc

- OMDoc: Open Math Documents [Koh06; Omd] models document & knowledge structures

level	**coverage**	**markup**
objects/phrases	presentation/content/text math	MathML, *OpenMath*, XHTML
statements	narrative, some declarations	OMDoc, XHTML
theories	domain theory graphs	OMDoc
documents	grouping	OMDoc
notations	*OpenMath* $\rightsquigarrow$ MathML rewrites	OMDoc
quiz, code, ...	ad-hoc	OMDoc

Background: Redesigning OMDoc

- OMDoc: Open Math Documents [Koh06; Omd] models document & knowledge structures

level	coverage	markup
objects/phrases	presentation/content/text math	MathML, *OpenMath*, XHTML
statements	narrative, some declarations	OMDoc, XHTML
theories	domain theory graphs	OMDoc
documents	grouping	OMDoc
notations	*OpenMath* $\rightsquigarrow$ MathML rewrites	OMDoc
quiz, code, ...	ad-hoc	OMDoc

- MMT: Meta Meta Theory [RK13; Uni] redesigns the formal core of OMDoc and gives it meaning

level	coverage	markup
objects	content math, literals	*OpenMath*+, MMT
statements	constant decls/defs	OMDoc, MMT
theories	domain/meta*	OMDoc, MMT
notations	one-level pres/parse	OMDoc, MMT
rules	typing, reconstruction, ...	Scala

Background: Redesigning OMDoc

- OMDoc: Open Math Documents [Koh06; Omd] models document & knowledge structures

level	coverage	markup
objects/phrases	presentation/content/text math	MathML, *OpenMath*, XHTML
statements	narrative, some declarations	OMDoc, XHTML
theories	domain theory graphs	OMDoc
documents	grouping	OMDoc
notations	*OpenMath* $\rightsquigarrow$ MathML rewrites	OMDoc
quiz, code, ...	ad-hoc	OMDoc

- MMT: Meta Meta Theory [RK13; Uni] redesigns the formal core of OMDoc and gives it meaning

level	coverage	markup
objects	content math, literals	*OpenMath*+, MMT
statements	constant decls/defs	OMDoc, MMT
theories	domain/meta*	OMDoc, MMT
notations	one-level pres/parse	OMDoc, MMT
rules	typing, reconstruction, ...	Scala

- **Definition 6.2.** iMMT: flexiformal MMT designs a formal language for the informal/narrative parts of OMDoc. (under development with Mihnea Iancu (diss))

Background: Redesigning OMDoc

- OMDoc: Open Math Documents [Koh06; Omd] models document & knowledge structures

level	coverage	markup
objects/phrases	presentation/content/text math	MathML, *OpenMath*, XHTML
statements	narrative, some declarations	OMDoc, XHTML
theories	domain theory graphs	OMDoc
documents	grouping	OMDoc
notations	*OpenMath* $\rightsquigarrow$ MathML rewrites	OMDoc
quiz, code, ...	ad-hoc	OMDoc

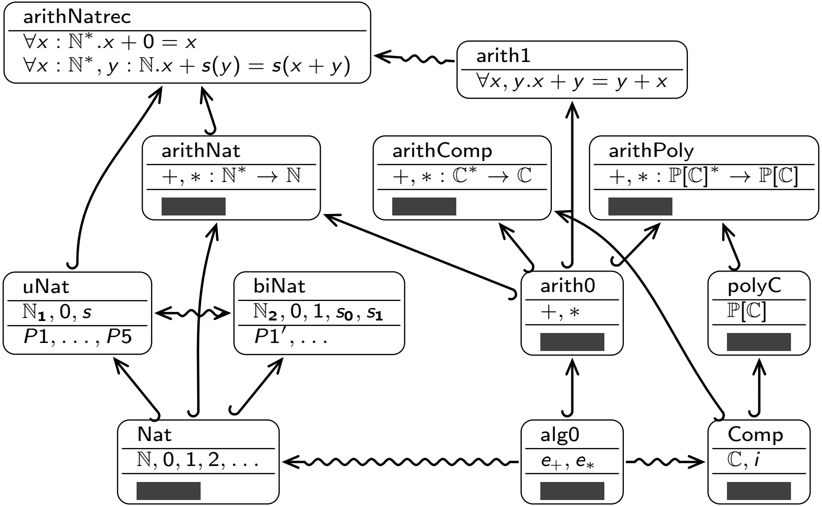
- MMT: Meta Meta Theory [RK13; Uni] redesigns the formal core of OMDoc and gives it meaning

level	coverage	markup
objects	content math, literals	*OpenMath*+, MMT
statements	constant decls/defs	OMDoc, MMT
theories	domain/meta*	OMDoc, MMT
notations	one-level pres/parse	OMDoc, MMT
rules	typing, reconstruction, ...	Scala

- **Definition 6.2.** iMMT: flexiformal MMT designs a formal language for the informal/narrative parts of OMDoc. (under development with Mihnea Iancu (diss))
- OMDoc2 $\widehat{=}$ MMT + iMMT (needs more language design)

Taking Informality Seriously in Theory Graphs

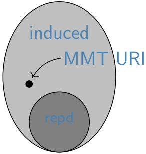
- Some of the contents are opaque to formal/syntactic methods



Using the MMT Copy Machine for Education

- **Recap:**

  Views/Structures are a giant copy-machine.
  The bushier the graph, the more induced content
  Invariant: every induced item as a canonical name,
  content can be regenerated from it.



- **Question:** Classically, only for formal content. Can we do that informally too?
- **Answer:** Yes we can, but
  - Template-based NL generation for view application, "β-reduction", definition expansion.
  - Symbolic NLP technology to to handle inflection, agreement, ...
- **Applications:**
  - guided tours, quiz/homework problems, and feedback for induced content.
  - refactoring for that

Using the MMT Copy Machine for Education (Example)

- The AI course at FAU introduces propositional logic as a domain description language for rational agents.

Definition 6.3. Let $\Sigma := \{\neg, \wedge, \Rightarrow, c_1, \ldots, c_n\}$ be a propositional signature, then the formulae of propositional logic are $\mathcal{L}_{\mathrm{PL}^0} := wfe(\Sigma)$.
We call $\mathcal{I}\colon \Sigma \to \bigcup_{n=0}^{\infty}(\mathbb{B}^n \to \mathbb{B})$ a model of propositional logic, iff $\mathcal{I}(f)\colon \mathbb{B}^k \to \mathbb{B}$ for any k-ary symbol $f \in \Sigma$ and

- $\mathcal{I}(\neg)(x) = \mathsf{T}$ iff $x = \mathsf{F}$,
- $\mathcal{I}(\wedge)(x, y) = \mathsf{T}$ iff $x = \mathsf{T}$ and $y = \mathsf{T}$, and
- $\mathcal{I}(\Rightarrow)(x, y) = \mathsf{T}$ iff $x = \mathsf{F}$ or $y = \mathsf{T}$.

We denote the set of models with $\mathcal{K}_{\mathrm{PL}^0}$.

- And the concept of satisfaction:

Definition 6.4. Let $\mathcal{I}$ be a model and A a formula of propositional logic. $\mathcal{I} \models_{\mathrm{PL}^0} \mathsf{A}$ iff $\mathcal{I}(\mathsf{A}) = \mathsf{T}$.
Definition 6.5. Let $\mathsf{A} \in \mathcal{L}_{\mathrm{PL}^0}$ be a formula.

- A model $\mathcal{I} \in \mathcal{K}_{\mathrm{PL}^0}$ satisfies A iff $\mathcal{I} \models_{\mathrm{PL}^0} \mathsf{A}$.
- A is satisfiable iff there exists a model that satisfies A.

Using the MMT Copy Machine for Education (Examples)

- In a Computational Logic course we introduce the abstract theory:

> **Definition 6.6.** A logical system (or simply a logic) is a triple $\mathcal{S} := \langle \mathcal{L}, \mathcal{M}, \models \rangle$, where
> 1. $\mathcal{L}$ is a set of propositions,
> 2. $\mathcal{M}$ a set of models, and
> 3. a relation $\models \subseteq \mathcal{M} \times \mathcal{L}$ called the satisfaction relation. We read $\mathcal{M} \models A$ as $\mathcal{M}$ satisfies A and correspondingly $\mathcal{M} \not\models A$ as $\mathcal{M}$ falsifies A.

- and prove that propositional logic is one. (a view)
- and/or use propositional logic as an example:

> **Example 6.7.** Propositional logic naturally forms a logical system $\langle \mathcal{L}_{\mathrm{PL^0}}, \mathcal{K}_{\mathrm{PL^0}}, \models_{\mathrm{PL^0}} \rangle$.

- **Recontextualization [KS24]:** In fact the latter can be generated from the former!

Using the MMT Copy Machine for Education (Problems)

- In the AI course we might be using the following problem:

Problem 6.1 (Satisfiability) Is the formula $c_1 \wedge (c_2 \Rightarrow \neg c_1)$ satisfiable? $\left[\ \boxtimes\ \text{Yes}\ \square\ \text{No}\ \right]$

- and give the following (very explicit) feedback for the wrong answer

Actually, there is a model $\mathcal{I}$ that satisfies $c_1 \wedge (c_2 \Rightarrow \neg c_1)$: it maps c_1 to T and c_2 to F. Then $\mathcal{I}(c_1 \wedge (c_2 \Rightarrow \neg c_1)) = \mathsf{T}$: Indeed, this can directly be seen by evaluating the truth table for $c_1 \wedge (c_2 \Rightarrow \neg c_1)$.

- **Idea 1:** Refactor this for logical systems (backwards over view above)
- **Idea 2:** For a propositional variable F, $\varphi : F \mapsto c_1 \wedge (c_2 \Rightarrow \neg c_1)$ is just a view!

Using the MMT Copy Machine for Education (Problems)

- **Recontextualization [KS24]:** Formulate the problem as

> **Problem 6.2**
> Is the formula F satisfiable? [☑ Yes ☐ No]
>
> *Feedback:* Actually, there is a model $\mathcal{M}$ that satisfies F: ⟨$\mathcal{M}$.definiens⟩ Then ⟨SatLogSys.conclusion⟩: ⟨SatLogSys.definiens⟩

and use everywhere via view chains (different logics/formulae)
Generates the feedback from an abstract (refactored) version as well:

> Actually, there is a model $\mathcal{I}$ that satisfies $c_1 \wedge (c_2 \Rightarrow \neg c_1)$: it maps c_1 to T and c_2 to F. Then $\mathcal{I}(c_1 \wedge (c_2 \Rightarrow \neg c_1)) = \mathrm{T}$: Indeed, this can directly be seen by evaluating the truth table for $c_1 \wedge (c_2 \Rightarrow \neg c_1)$.

Implementations of Relocalization

- **Relocalization at the sTEX level:** sTEX schemata to be filled with view components [KS24].
- **Problems & Advantages:**
  - – sometimes un-grammatical, often clumsy/circuitous language generated
  - \+ Generated sTEX sources can be hand optimized
  - – generated material must go through the sTEX/RusTEX/FLAMS pipeline.

Implementations of Relocalization

- **Relocalization at the sTEX level:** sTEX schemata to be filled with view components [KS24].
- **Problems & Advantages:**
  - – sometimes un-grammatical, often clumsy/circuitous language generated
  - + Generated sTEX sources can be hand optimized
  - – generated material must go through the sTEX/RusTEX/FLAMS pipeline.
- **Relocalization at the GF AST level:** cf. [CICM25]
  1. Parse relocalizable material and view assignments into AST via GF.
  2. View application by AST-to-AST replacement/reduction.
  3. Simplification via AST-to-AST rewriting. (collected at `https://github.com/flexiformal/rewriting-rules`)
- **Problems & Advantages:**
  - + Nice grammatical, streamlined language
  - +/– Generated FTML format cannot/need not be hand optimized
  - – Still need much better grammar coverage (how semantic should it be?)
  - – Still need many more simplification rules (but they seem to be foundational/canonical)

Simplifying Paths in a Graph

- **Example 6.8.** We relocalize the definition of a path from graphs to NFAs.

Definition In an NFA, a *path* is a finite sequence $t_1, \ldots, t_n$ of transitions $t_i := \langle q_i, c_i, {q'}_i \rangle$ with ${q'}_i = q_{i+1}$ for all $1 \leq i < n$.

Derived From

> **Definition** In a graph, a *path* is a finite sequence $e_1, \ldots, e_n$ of edges with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.

By Applying

> **Theorem** An NFA $\langle Q, \Sigma, \delta, q_0, F \rangle$ admits a graph $\langle V, E, s, t, L, l \rangle$, where $V := Q$, $E := \{t \,|\, t \text{ is a transition}\}$, $s := \pi_1$, $t := \pi_3$, $L := \Sigma$, and $l := \pi_2$.

Simplifying Paths in a Graph

- **Example 6.8.** We relocalize the definition of a path from graphs to NFAs.

1. **Comprehension term reduction:** First we reduce the comprehension term expression "*elements of* $\{t \mid t$ *is a transition*$\}$" to "*transitions*":

   > A *path* is a finite sequence $e_1, \ldots, e_n$ of transitions with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

2. **Structure expansion:** Then we expand the noun phrase "*finite sequence* $e_1, \ldots, e_n$ *of transitions*" by appending a variable definition "$e_k := \langle q_k, c_k, {q'}_k \rangle$":

   > A *path* is a finite sequence $e_1, \ldots, e_n$ of transitions $e_k := \langle q_k, c_k, {q'}_k \rangle$ with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

3. **Variable expansion:** This allows us to expand later occurrences of "e" ("e_i" and "e_{i+1}"):

   > A *path* is a finite sequence $e_1, \ldots, e_n$ of transitions $e_k := \langle q_k, c_k, {q'}_k \rangle$ with $\pi_3(\langle q_i, c_i, {q'}_i \rangle) = \pi_1(\langle q_{i+1}, c_{i+1}, {q'}_{i+1} \rangle)$ for all $1 \leq i < n$.

4. **Projection reduction:** Now that the arguments of the projections are triples we can evaluate the projections:

   > A *path* is a finite sequence $e_1, \ldots, e_n$ of transitions $e_k := \langle q_k, c_k, {q'}_k \rangle$ with ${q'}_i = q_{i+1}$ for all $1 \leq i < n$.

5. **Variable renaming (optional):** As a last step we rename the variables e_j to t_j:

   > A *path* is a finite sequence $t_1, \ldots, t_n$ of transitions $t_k := \langle q_k, c_k, {q'}_k \rangle$ with ${q'}_i = q_{i+1}$ for all $1 \leq i < n$.

But it can become even more opaque

- In my world (of theory graphs) flexiformality can appear even earlier.
- In flexiformal views even the proof obligations can be unspecified.
- **Example 6.9.** Consider the following two theories

> **Theory: Normed_VS** If $\mathcal{V}$ is a vector space over F, then $|\cdot| : \mathcal{F} \to F$ is called a **norm**, iff for all $a \in F$ and $u, v \in V$ we have (A) $|av| = |a|\ |v|$, (T) $|u+v| \leq |u| + |v|$, and (S) $v = 0$ if $|v| = 0$.

> **Theory: Metric_Space** Let M be a set, then we call a function $d : M^2 \to \mathbb{R}$ a **metric** on M, iff (I) $d(x,y) = 0$ iff $x = y$, (S) $d(x,y) = d(y,x)$, and (T) $d(x,z) \leq d(x,y) + d(y,z)$.

- **Example 6.10 (Empty View).**

> **View: Metric_Space** $\to$ **Normed_VS** this is well-known.

But it can become even more opaque

- In my world (of theory graphs) flexiformality can appear even earlier.
- In flexiformal views even the proof obligations can be unspecified.
- **Example 6.9.** Consider the following two theories

Theory: Normed_VS If $\mathcal{V}$ is a vector space over F, then $|\cdot| : \mathcal{F} \to F$ is called a **norm**, iff for all $a \in F$ and $u, v \in V$ we have (A) $|av| = |a|\,|v|$, (T) $|u+v| \leq |u| + |v|$, and (S) $v = 0$ if $|v| = 0$.

Theory: Metric_Space Let M be a set, then we call a function $d : M^2 \to \mathbb{R}$ a **metric** on M, iff (I) $d(x,y) = 0$ iff $x = y$, (S) $d(x,y) = d(y,x)$, and (T) $d(x,z) \leq d(x,y) + d(y,z)$.

Example 6.11 (Opaque View with partial symbol mapping).

View: Metric_Space → Normed_VS $M \mapsto \mathcal{V}$, $d(x,y) \mapsto |x-y|$

But it can become even more opaque

- In my world (of theory graphs) flexiformality can appear even earlier.
- In flexiformal views even the proof obligations can be unspecified.
- **Example 6.9.** Consider the following two theories

> **Theory: Normed_VS** If $\mathcal{V}$ is a vector space over F, then $|\cdot| : \mathcal{F} \to F$ is called a **norm**, iff for all $a \in F$ and $u, v \in V$ we have (A) $|av| = |a|\,|v|$, (T) $|u + v| \leq |u| + |v|$, and (S) $v = 0$ if $|v| = 0$.

> **Theory: Metric_Space** Let M be a set, then we call a function $d : M^2 \to \mathbb{R}$ a **metric** on M, iff (I) $d(x, y) = 0$ iff $x = y$, (S) $d(x, y) = d(y, x)$, and (T) $d(x, z) \leq d(x, y) + d(y, z)$.

Example 6.12 (Opaque View with proof obligations).

> **View: Metric_Space $\to$ Normed_VS** $M \mapsto \mathcal{V}$, $d(x, y) \mapsto |x - y|$, $I \mapsto$ A $S \mapsto$ S , $T \mapsto$ T

But it can become even more opaque

- In my world (of theory graphs) flexiformality can appear even earlier.
- In flexiformal views even the proof obligations can be unspecified.
- **Example 6.9.** Consider the following two theories

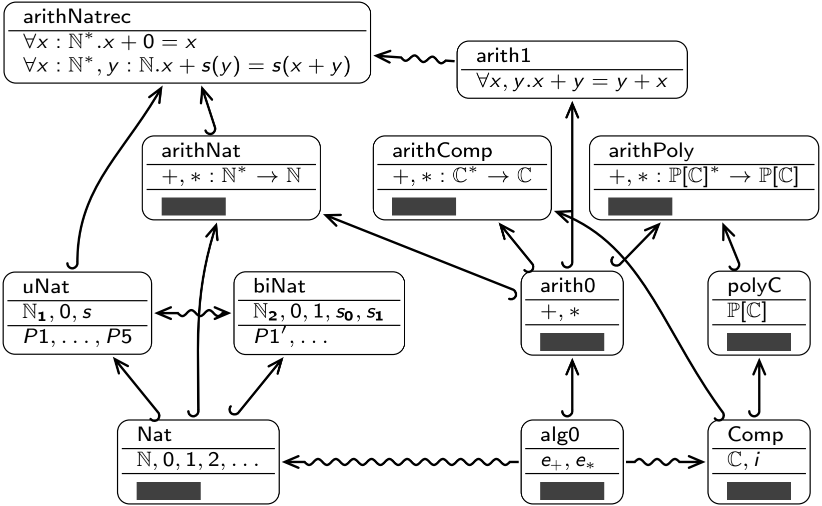
> **Theory: Normed_VS** If $\mathcal{V}$ is a vector space over F, then $|\cdot| : \mathcal{F} \to F$ is called a **norm**, iff for all $a \in F$ and $u, v \in V$ we have (A) $|av| = |a|\,|v|$, (T) $|u+v| \leq |u| + |v|$, and (S) $v = 0$ if $|v| = 0$.

> **Theory: Metric_Space** Let M be a set, then we call a function $d : M^2 \to \mathbb{R}$ a **metric** on M, iff (I) $d(x,y) = 0$ iff $x = y$, (S) $d(x,y) = d(y,x)$, and (T) $d(x,z) \leq d(x,y) + d(y,z)$.

Example 6.13 (Formal View). With full proof terms.

Taking Informality Seriously in Theory Graphs

- Some of the contents are opaque to formal/syntactic methods



Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\leadsto$ symbolic core)

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALEA)
  - Automation of flexiformalization is possible/desirable ($\leadsto$ symbolic core)
- **Flexiformalization:**
  - Relax on "verification", gain machine-actionable artifacts.
  - Informal/formal continuum allows incremental formalization.
  - We can keep modularization and proofs for automation. (maybe opaque)
  - Flexiformal representation formats like sTEX mix formal/informal parts.

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\rightsquigarrow$ symbolic core)
- **Flexiformalization:**
- **Applications:** Incremental formalization/informalization, active documents, education,...

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\leadsto$ symbolic core)
- **Flexiformalization:**
- **Applications:** Incremental formalization/informalization, active documents, education,. . .
- **Flexiformal Libraries/Workflows:**
  - Focus on building (small) flexiformal artifacts ($\widehat{=}$ elaboration?)
  - TEX parsing, macro expansion takes 95% of the build time. (pdflatex/rusTeX)
  - Separate compilation and document contextualization as a solution?

Conclusions & Future Work

- **Take Home Messages:** (flexiformal annotation)
  - There is a way of dealing with math language beyond NLU and NLG, and CNL (or LLMs).
  - Currently we use biological periphery (i.e. humans) for flexiformalization and language design. (towards a foundation of informal mathematics)
  - Test this by fielding semantic support services (currently7 ALeA)
  - Automation of flexiformalization is possible/desirable ($\rightsquigarrow$ symbolic core)
- **Flexiformalization:**
- **Applications:** Incremental formalization/informalization, active documents, education,...
- **Flexiformal Libraries/Workflows:**
- **Ongoing/Future Work:**
  - Harvest a lexical resource MathLex: `https://github.com/OpenMath/mathlex`
  - Use the theory graph structure for re-usability
  - FAIR (Findable, Accessible, Reusable, Interoperable) Math
  - A flexiformal domain model for undergraduate Math/CS ($\widehat{=}$ MathLib$^-$)
  - Semantic services (learning interventions) for ALeA (`https://alea.education`)