Corpus-Based Meta-Mathematics: Flexiformal Mathematics across various Systems and Languages

Michael Kohlhase

Professur für Wissensrepräsentation und -verarbeitung
Informatik, FAU Erlangen-Nürnberg
http://kwarc.info

February 4. 2020, Hausdorff-Center, Bonn

1 Intro: What Peter Asked me to talk about "formal mathematics across various systems and languages"

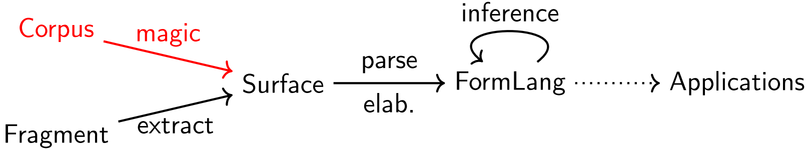
The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications

$$
\begin{tikzcd}
\text{Corpus} \arrow[dr, "\text{magic}"] & & & \\
& \text{Surface} \arrow[r, "\text{parse}", "\text{elab.}"'] & \text{FormLang} \arrow[loop above, "\text{inference}"] \arrow[r, dotted] & \text{Applications} \\
\text{Fragment} \arrow[ur, "\text{extract}"'] & & &
\end{tikzcd}
$$

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\leadsto$ applications

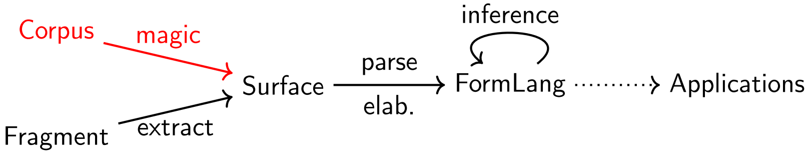


Project	Corpus	Surface	Formal Language	Appl./Systems
Flyspec	Book@NL	HolLight	HOL@OCaml	Verification
FAbstracts1	NL-Def/Thm	Lean	CIC	Domain Documentation/Stadardization

- See Tom's talk on the first day.

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications



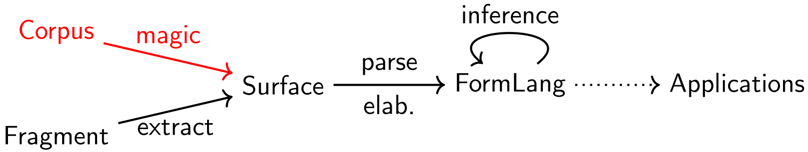
-

Project	Corpus	Surface	Formal Language	Appl./Systems
Naproche	MNL	CNL(FOL)	DFG@EProver	Verification
SAD	MNL	ForThel	FOL@SAD	Verification
Naproche-SAD	MNL	ForThel	Isabelle	Verification
Diproche	textbook	DiThel?	(Anti-)ATP	Learning Feedback
FAbstracts2	NL-Def/Thm	CNL(T)	CIC	Domain Documentation/Stadardization

- See the talks of Peter/Tom/Naproche Team

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications



Project	Corpus	Surface	Formal Language	Appl./Systems
Coq-Export	Coq-Libs	CoqXML	MMT/CIC	Reuse /Search
GAP-Export	GAPLib/Doc	GAP-JSON	MMT/GAP	Interoperability

- Work of the KWARC group in the OAF/OpenDreamKit projects

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications

$$\begin{tikzcd} \text{Corpus} \arrow[dr, "\text{magic}"] & & & \\ & \text{Surface} \arrow[r, "\text{parse}", "\text{elab.}"'] & \text{FormLang} \arrow[loop above, "\text{inference}"] \arrow[r, dotted] & \text{Applications} \\ \text{Fragment} \arrow[ur, "\text{extract}"'] & & & \end{tikzcd}$$

-

Project	Corpus	Surface	Formal Language	Appl./Systems
sTeX	NL-Courses	sTeX	OMDoc	Smart Courseware & cf. FAbstracts
SMGloM	NL-Defs	sTeX	OMDoc	Glossary/Lexicon/Dictionary

- Long-term KWARC infrastructure projects

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications

$$
\begin{tikzcd}
\text{Corpus} \arrow[dr, red, "\text{magic}"] & & & \\
& \text{Surface} \arrow[r, "\text{parse}", "\text{elab.}"'] & \text{FormLang} \arrow[loop above, "\text{inference}"] \arrow[r, dotted] & \text{Applications} \\
\text{Fragment} \arrow[ur, "\text{extract}"'] & & &
\end{tikzcd}
$$

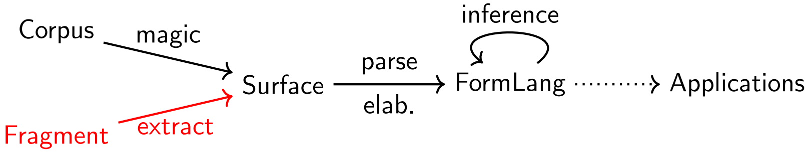
-

Project	Fragment	Surface	Formal Language	Appl./Systems
HOTT	HOTT-Book	GF-AST	GF-AST	Translation

- See Aarne's Talk

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications



-

Project	Fragment	Surface	Formal Language	Appl./Systems
∂GF	NL	GF-Islands	?????	Concordances
QE-Extraction	NL	OpenMath	Unit Algebra	Search/Accessibility/Conversio

- A M.Sc. project at FAU: convert $3m/s$ to "meters per second", but not $E = mc^2$

The Flexi-Formalization Pipeline Across Systems/Projects

- The math flexiformalization process: corpus $\rightsquigarrow$ applications

$$\begin{tikzcd}
\text{Corpus} \arrow[dr, "\text{magic}"] & & & \\
& \text{Surface} \arrow[r, "\text{parse}", "\text{elab.}"'] & \text{FormLang} \arrow[loop above, "\text{inference}"] \arrow[r, dotted] & \text{Applications} \\
\text{Fragment} \arrow[ur, red, "\text{extract}"'] & & &
\end{tikzcd}$$

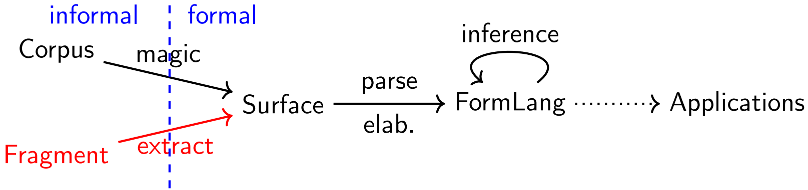
-

Project	Fragment	Surface	Formal Language	Appl./Systems
GLF	NL	GF-AST	MitM	Translation

- A declarative formalization pipeline using GF, MMT, and λ-ProLog (see below)

The Flexi-Formalization Pipeline Across Systems/Projects

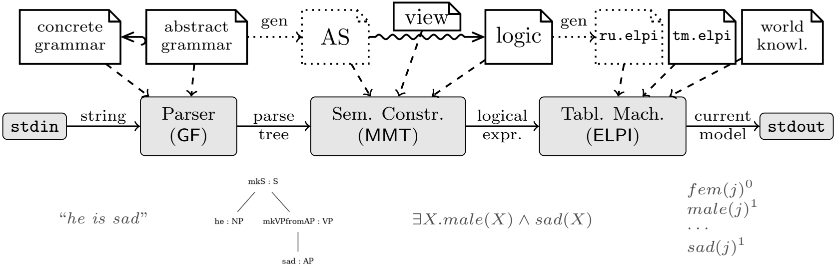
- The math flexiformalization process: corpus $\rightsquigarrow$ applications



Project	Fragment	Surface	Formal Language	Appl./Systems
GLF	NL	GF-AST	MitM	Translation
- **Definition 1.1.** informal $\widehat{=}$ not formal, formalization $\widehat{=}$ make formal!

A declarative Formalization Pipeline (cf. Aarne)

- Bring your own Logic: The following formalization pipeline is logic-independent.



To change the target logic+calculus, just write another one in LF+X.

What is it we do? Corpus-Based Meta-Mathematics

- **Mathematics**: as a test tube for STEM
  - the knowledge and document structures are quite explicit and overt
  - the content of mathematics is well-understood.

  Consider anything that has the same properties as "mathematics" as well.

What is it we do? Corpus-Based Meta-Mathematics

- **Mathematics**: as a test tube for STEM
  - the knowledge and document structures are quite explicit and overt
  - the content of mathematics is well-understood.

  Consider anything that has the same properties as "mathematics" as well.
- **Meta**: we develop meta-artefacts, i.e. we
  - design **representation languages** (logics) that allow to talk about mathematical objects, their properties, and relations,
  - invent **algorithms** that analyze and transform these representations, and
  - implement them in **end-user systems** that utilize both.

What is it we do? Corpus-Based Meta-Mathematics

- **Mathematics**: as a test tube for STEM
  - the knowledge and document structures are quite explicit and overt
  - the content of mathematics is well-understood.

  Consider anything that has the same properties as "mathematics" as well.
- **Meta**: we develop meta-artefacts, i.e. we
  - design **representation languages** (logics) that allow to talk about mathematical objects, their properties, and relations,
  - invent **algorithms** that analyze and transform these representations, and
  - implement them in **end-user systems** that utilize both.
- **Corpus-based**: we do this as a natural science by looking at data (i.e. corpora of documents and formalizations).

What is it we do? Corpus-Based Meta-Mathematics

- **Mathematics**: as a test tube for STEM
  - the knowledge and document structures are quite explicit and overt
  - the content of mathematics is well-understood.

  Consider anything that has the same properties as "mathematics" as well.
- **Meta**: we develop meta-artefacts, i.e. we
  - design **representation languages** (logics) that allow to talk about mathematical objects, their properties, and relations,
  - invent **algorithms** that analyze and transform these representations, and
  - implement them in **end-user systems** that utilize both.
- **Corpus-based**: we do this as a natural science by looking at data (i.e. corpora of documents and formalizations).
- **Process**: approach corpus-based meta-mathematics (iteratively) in three steps:
  - **Analysis**: we analyze the corpora for patterns and structures.
  - **Synthesis**: we design and build meta-artefacts (languages, algorithms, and systems) and derive data sets from the corpora.
  - **Experimentation**: we evaluate the representation languages and algorithms on the corpora and the systems on end users (mathematicians).

2 Flexiformality

Formalization in Mathematical Practice

- To formalize maths in a formal system $\mathcal{S}$, we need to choose a foundation, i.e. a foundational $\mathcal{S}$-theory, e.g. a set theory like ZFC.
- Formality is an all-or-nothing property (a single "obviously" can ruin it.)
- Almost all mathematical documents are informal in 4 ways:
  - the foundation is unspecified (they are essentially equivalent)
  - the language is informal (essentially opaque to MKM algos.)
  - even formulae are informal (presentation markup)
  - context references are underspecified
    - mathematical objects and concepts are often identified by name
    - statements (citations of definitions, theorems, and proofs) underspecified
    - theories and theory reuse not marked up at all
- The gold standard of mathematical communication is "rigor" (cf. [BC01])

Formalization in Mathematical Practice

- To formalize maths in a formal system $\mathcal{S}$, we need to choose a foundation, i.e. a foundational $\mathcal{S}$-theory, e.g. a set theory like ZFC.
- Formality is an all-or-nothing property (a single "obviously" can ruin it.)
- Almost all mathematical documents are informal in 4 ways:
- The gold standard of mathematical communication is "rigor" (cf. [BC01])
  - **Definition 2.1.** We call a mathematical document rigorous, if it could be formalized in a formal system given enough resources.
  - This possibility is almost always unconsummated
  - Why?: There are four factors that disincentivize formalization for Maths

    propaganda: *Maths is done with pen and paper*

    tedium: de Bruijn factors ~ 4 for current systems (details in [Wie12])

    inflexibility: formalization requires commitment to formal system and foundation

    proof verification useless: peer reviewing works just fine for Math
  - **Definition 2.2.** The de Bruijn factor is the quotient of the lengths of the formalization and the original text.

Formalization in Mathematical Practice

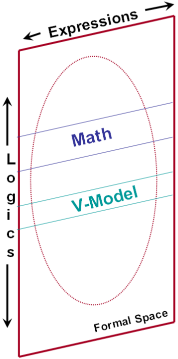
- To formalize maths in a formal system $\mathcal{S}$, we need to choose a foundation, i.e. a foundational $\mathcal{S}$-theory, e.g. a set theory like ZFC.
- Formality is an all-or-nothing property (a single "obviously" can ruin it.)
- Almost all mathematical documents are informal in 4 ways:
- The gold standard of mathematical communication is "rigor" (cf. [BC01])
  - **Definition 2.1.** We call a mathematical document rigorous, if it could be formalized in a formal system given enough resources.
  - This possibility is almost always unconsummated
  - Why?: There are four factors that disincentivize formalization for Maths
    propaganda: *Maths is done with pen and paper*
    tedium: de Bruijn factors ~ 4 for current systems (details in [Wie12])
    inflexibility: formalization requires commitment to formal system and foundation
    proof verification useless: peer reviewing works just fine for Math
  - **Definition 2.2.** The de Bruijn factor is the quotient of the lengths of the formalization and the original text.

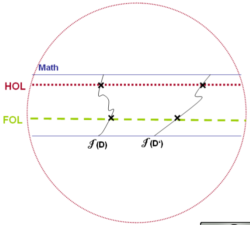
  In Effect: Hilbert's program has been comforting but useless

- Question: What can we do to change this?

What is Informal Mathematical Knowledge

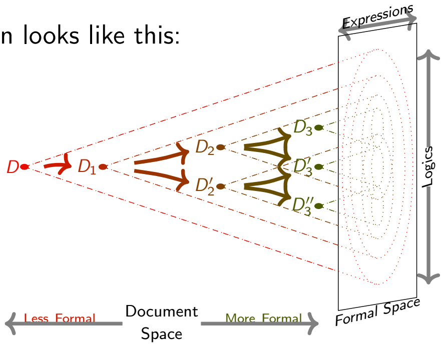
- Idea: informal knowledge could be formalized (but isn't yet!)
- **Definition 2.3.** The meaning of a knowledge item is the set of all its formalizations
- Problem: What is the space of formalizations?
- **Definition 2.4.** The formal space is the set $\mathcal{F} := \{\langle S, e \rangle \mid S \in \mathfrak{F}, e \in \mathcal{L}(S)\}$, where $\mathfrak{F}$ is the class of formal systems and $\mathcal{L}(S)$ is the language of S. (i.e. every formal expression is a point in $\mathcal{F}$)
- Different Logics correspond to different bands
- The meaning of $\mathcal{D}$ is a set $\mathcal{I}(\mathcal{D}) \subseteq \mathcal{F}$.
- $\mathcal{D}$ can be formalized in multiple logics $\mathcal{I}(\mathcal{D})$ forms a cross-section of logic-bands.





A Formality Ordering on $\mathcal{F}$

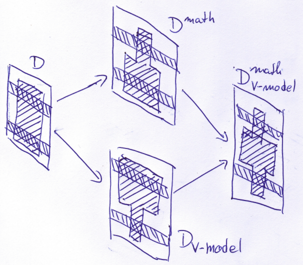
- Stepwise formalization looks like this:



- **Definition 2.5.** $\mathcal{D}$ is more formal than $\mathcal{D}'$ (write $\mathcal{D} \lll \mathcal{D}'$), iff $\mathcal{I}(\mathcal{D}) \subset \mathcal{I}(\mathcal{D}')$.
- This partial ordering relation answers the question of "graded formality" or the nature of "stepwise formalization" raised above.

Stepwise Formalization in Multiple Dimensions

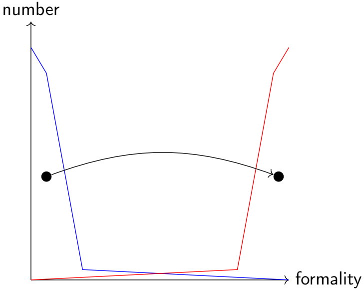
- Empirically: Formalization is a stepwise process of (order of steps may vary)
  - spotting semantic objects (from the surrounding text)
  - chunking: grouping them for re-use (e.g. assigning to home theories)
  - relating: making their relationships explicit (this is used by semantic services)
- In multi-dimensional situations:



  - any formalization step on $\mathcal{D}$ trims $\mathcal{I}(\mathcal{D})$.
  - not all "steps" are comparable in $\lll$
  - but per-dimension formalization is confluent
- Observation: This is the normal situation, we coin a new concept to describe it.
- **Definition 2.6.** We call a representation flexiform, iff it is of flexible formality in any of the adequate dimensions of formality.

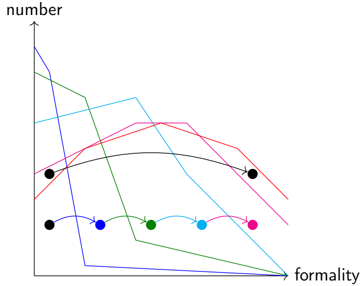
Migration by Stepwise Formalization

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



Migration by Stepwise Formalization

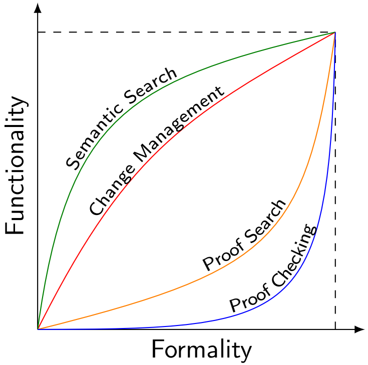
- Full Formalization is hard (we have to commit, make explicit)
- Let's look at documents and document collections.
- Partial formalization allows us to
  - formalize stepwise, and
  - be flexible about the depth of formalization.



Functionality of Flexiformal Services

- Generally: Flexiformal services deliver according to formality level (GIGO: Garbage in $\rightsquigarrow$ 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)

References I

- Serge Autexier and Claudio Sacerdoti Coen. "A Formal Correspondence Between OMDoc with Alternative Proofs and the $\overline{\lambda}\mu\tilde{\mu}$-calculus". In: *Mathematical Knowledge Management (MKM)*. Ed. by Jon Borwein and William M. Farmer. LNAI 4108. Springer Verlag, 2006, pp. 67–81. DOI: 10.1007/11812289_7.
- Henk Barendregt and Arjeh M. Cohen. "Electronic communication of mathematics and the interaction of computer algebra systems and proof assistants". In: *Journal of Symbolic Computation* 32 (2001), pp. 3–22.
- Michael Kohlhase. "The Flexiformalist Manifesto". In: *14th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*. Ed. by Andrei Voronkov et al. Timisoara, Romania: IEEE Press, 2013, pp. 30–36. ISBN: 978-1-4673-5026-6. URL: http://kwarc.info/kohlhase/papers/synasc13.pdf.

References II

Christoph Lange and Michael Kohlhase. "A Mathematical Approach to Ontology Authoring and Documentation". In: *MKM/Calculemus Proceedings*. Ed. by Jacques Carette et al. LNAI 5625. Springer Verlag, July 2009, pp. 389–404. ISBN: 978-3-642-02613-3. URL: http://kwarc.info/kohlhase/papers/mkm09-omdoc4onto.pdf.

Freek Wiedijk. *The "de Bruijn factor"*. web page at http://www.cs.ru.nl/~freek/factor/. Mar. 1, 2012. URL: http://www.cs.ru.nl/~freek/factor/.