Knowledge Representation for Mathematical/Technical Knowledge
Summer Semester 2018

– Provisional Lecture Notes –

Prof. Dr. Michael Kohlhase & Dennis Müller, M.Sc.
Professur für Wissensrepräsentation und -verarbeitung
Informatik, FAU Erlangen-Nürnberg
Michael.Kohlhase,Dennis.Mueller@FAU.de

April 18, 2018

Preface

Course Concept

Aims: To give students a solid foundation of the basic concepts and practices in representing mathematical/technical knowledge, so they can do (guided) research in the KWARC group.

Organization: Theory and Practice: The KRMT course intended to give a small cohort of students (≤ 15) the opportunity to understand theoretical and practical aspects of knowledge representation for technical documents. The first aspect will be taught as a conventional lecture on computational logic (focusing on the expressive formalisms needed account for the complexity of mathematical objects) and the second will be served by the "KRMT Lab", where we will jointly (instructors and students) develop representations for technical documents and knowledge. Both parts will roughly have equal weight and will alternate weekly.

Prerequisites: The course builds on the logic courses in the FAU Bachelor's program, in particular the course "Grundlagen der Logik in der Informatik" (GLOIN). While prior exposure to logic and inference systems e.g. in GLOIN or the AI-1 course is certainly advantageous to keep up, it is not strictly necessary, as the course introduces all necessary prerequisites as we go along. So a strong motivation or exposure to strong abstraction and mathematical rigour in other areas should be sufficient.

Similarly, we do not presuppose any concrete mathematical knowledge – we mostly use (very) elementary algebra as example domain – but again, exposure to proof-based mathematical practice – whatever it may be – helps a lot.

Course Contents and Organization

The course concentrates on the theory and practice of representing mathematical knowledge in a wide array of mathematical software systems.

In the theoretical part we concentrate on computational logic and mathematical foundations; the course notes are in this document. In the practical part we develop representations of concrete mathematical knowledge in the MMT system, unveiling the functionality of the system step by step. This process is tracked in a tutorial separate document [OMT].

Excursions: As this course is predominantly about modeling natural language and not about the theoretical aspects of the logics themselves, we give the discussion about these as a "suggested readings" ?sec?. This material can safely be skipped (thus it is in the appendix), but contains the missing parts of the âĂIJbridgeâĂİ from logical forms to truth conditions and textual entailment.

This Document

This document contains the course notes for the course "Knowledge Representation for Mathematical/Technical Knowledge" ("Logik-Basierte WissensreprÃďsentation fÃijr Mathematisch/Technisches Wissen") in the Summer Semesters 17 ff.

Format: The document mixes the slides presented in class with comments of the instructor to give students a more complete background reference.

Caveat: This document is made available for the students of this course only. It is still very much a draft and will develop over the course of the current course and in coming academic years.

Knowledge Representation Experiment:

This document is also an experiment in knowledge representation. Under the hood, it uses the sTeX package [Koh08; Koh17], a TeX/LaTeX extension for semantic markup, which allows to export the contents into active documents that adapt to the reader and can be instrumented with services based on the explicitly represented meaning of the documents.

Comments: and extensions are always welcome, please send them to the author.

Other Resources: The course notes are complemented by a tutorial on formalization mathematical Knowledge in the MMT system [OMT] and the formalizations at `https://gl.mathhub.info/Tutorials/Mathematicians`.

Acknowledgments

Materials: All course materials have bee restructured and semantically annotated in the sTeX format, so that we can base additional semantic services on them (see slide 6 for details).

CompLog Students: The course is based on a series of courses "Computational Logic" held at Jacobs University Bremen and shares a lot of material with these. The following students have submitted corrections and suggestions to this and earlier versions of the notes: Rares Ambrus, Florian Rabe, Deyan Ginev, Fulya Horozal, Xu He, Enxhell Luzhnica, and Mihnea Iancu.

KRMT Students: The following students have submitted corrections and suggestions to this and earlier versions of the notes: Michael Banken

Recorded Syllabus for SS 2018

In this document, we record the progress of the course in the summer semester 2018 in the form of a "recorded syllabus", i.e. a syllabus that is created after the fact rather than before.

Recorded Syllabus Summer Semester 2018:

#	date	until	slide	page

Here the syllabus of the last academic year for reference, the current year should be similar; see the course notes of last year available for reference at http://kwarc.info/teaching/KRMT/notes-SS17.pdf.

Recorded Syllabus Summer Semester 2017:

#	date	until	slide	page
1	4. May	overview, some admin, math search		
2	8. May	framing, theory graphs,content/form		
3	11. May	$\mathbb{N}, +$ in MMT		

Contents

Chapter 1

Administrativa

We will now go through the ground rules for the course. This is a kind of a social contract between the instructor and the students. Both have to keep their side of the deal to make learning as efficient and painless as possible.

Prerequisites

- ▷ the mandatory courses from Semester 1-4, in particular: (or equivalent)
  - ▷ course "Grundlagen der Logik in der Informatik" (GLOIN)
  - ▷ CS Math courses "Mathematik C1-4" (IngMath1-4) (our "domain")
  - ▷ algorithms and data structures
  - ▷ course "Künstliche Intelligenz I" (nice-to-have only)
- ▷ Motivation, Interest, Curiosity, hard work
  - ▷ You can do this course if you want! (and we will help you)

 1

Now we come to a topic that is always interesting to the students: the grading scheme.

Grades

- ▷ Academic Assessment: two parts (Portfolio Assessment)
  - ▷ 20-min oral exam at the end of the semester (50%)
  - ▷ results of the KRMT lab (50%)

 2

KRMT Lab (Dogfooding our own Techniques)

- ▷ (generally) we use the thursday slot to get our hands dirty with actual representations.

- Instructor: Dennis Müller (dennis.mueller@fau.de) Room: 11.138, Tel: 85-64053
- Goal: Reinforce what was taught in class and have some fun
- Homeworks: will be small individual problem/programming/proof assignments (but take time to solve) group submission if and only if explicitly permitted
- Admin: To keep things running smoothly
  - Homeworks will be posted on course forum (discussed in the lab)
  - No "submission", but open development on a git repos. (details follow)
- Homework Discipline:
  - start early! (many assignments need more than one evening's work)
  - Don't start by sitting at a blank screen
  - Humans will be trying to understand the text/code/math when grading it.

Textbook, Handouts and Information, Forums

- (No) Textbook: Course notes will be posted at http://kwarc.info/teaching/KRMT
  - I mostly prepare them as we go along (semantically preloaded $\leadsto$ research resource)
  - please e-mail me any errors/shortcomings you notice. (improve for the group)
- Announcements will be posted on the course forum
  - https://fsi.cs.fau.de/forum/150-Logikbasierte-Wissensrepraesentation
- Check the forum frequently for
  - announcements, homeworks, questions
  - discussion among your fellow students

Do I need to attend the lectures

- Attendance is not mandatory for the KRMT lecture (official version)
- There are two ways of learning: (both are OK, your mileage may vary)
  - Approach B: Read a book/papers
  - Approach I: come to the lectures, be involved, interrupt me whenever you have a question.

The only advantage of I over B is that books/papers do not answer questions

- Approach S: come to the lectures and sleep does not work!
- The closer you get to research, the more we need to discuss!

Next we come to a special project that is going on in parallel to teaching the course. I am using the course materials as a research object as well. This gives you an additional resource, but may affect the shape of the coures materials (which now serve double purpose). Of course I can use all the help on the research project I can get, so please give me feedback, report errors and shortcomings, and suggest improvements.

Experiment: E-Learning with KWARC Technologies

- My research area: deep representation formats for (mathematical) knowledge
- Application: E-learning systems (represent knowledge to transport it)
- Experiment: Start with this course (Drink my own medicine)
  - Re-Represent the slide materials in *OMDoc* (Open Math Documents)
  - Feed it into the PantaRhei system (http://panta.kwarc.info)
  - Try it on you all (to get feedback from you)
- Tasks (Unfortunately, I cannot pay you for this; maybe later)
  - help me complete the material on the slides (what is missing/would help?)
  - I need to remember "what I say", examples on the board. (take notes)
- Benefits for you (so why should you help?)
  - you will be mentioned in the acknowledgements (for all that is worth)
  - you will help build better course materials (think of next-year's students)

Chapter 2

Overview over the Course

Plot of this Course

- ▷ Today: Motivation, Admin, and find out what you already know
  - ▷ What is logic, knowledge representation
  - ▷ What is mathematical/technical knowledge
  - ▷ how can you get involved with research at KWARC

2.1 Introduction & Motivation

Knowledge-Representation and -Processing

- ▷ **Definition 2.1.1 (True and Justified Belief)** Knowledge is a body of facts, theories, and rules available to persons or groups that are so well justified that their validity/truth is assumed.
- ▷ **Definition 2.1.2** Knowledge representation formulates knowledge in a formal language so that new knowledge can be induced by inferred via rule systems (inference).
- ▷ **Definition 2.1.3** We call an information system knowledge-based, if a large part of its behaviour is based on inference on represented knowledge.
- ▷ **Definition 2.1.4** The field of knowledge processing studies knowledge-based systems, in particular
  - ▷ compilation and structuring of explicit/implicit knowledge (knowledge acquisition)
  - ▷ formalization and mapping to realization in computers (knowledge representation)
  - ▷ processing for problem solving (inference)
  - ▷ presentation of knowledge (informationvisualization)

▷ knowledge representation and processing are subfields of symbolic artificial intelligence

Mathematical Knowledge (Representation and -Processing)

▷ KWARC (my research group) develops foundations, methods, and applications for the representation and processing of mathematical knowledge

- ▷ Mathematics plays a fundamental role in Science and Technology(practice with maths, apply in STEM)
- ▷ mathematical knowledge is rich in content, sophisticated in structure, and explicitly represented ...
- ▷ ..., and we know exactly what we are talking about (in contrast to economics or love)

Working Definition: Everything we understand well is "mathematics" (e.g. CS, Physics, ...)

▷ There is a lot of mathematical knowledge

- ▷ 120,000 Articles are published in pure/applied mathematics (3.5 millions so far)
- ▷ 50 Millionen science articles in 2010 [Jin10] with a doubling time of 8-15 years [LI10]
- ▷ 1 M Technical Reports on http://ntrs.nasa.gov/ (e.g. the Apollo reports)
- ▷ a Boeing-Ingenieur tells of a similar collection (but in Word 3,4,5,...)

About Humans and Computers in Mathematics

▷ Computers and Humans have complementary strengths.

- ▷ Computers can handle large data and computations flawlessly at enormous speeds.
- ▷ Humans can sense the environment, react to unforeseen circumstances and use their intuitions to guide them through only partially understood situations.

In mathematics: we exploit this, we

▷
- ▷ let humans explore mathematical theories and come up with novel insights/proofs,
- ▷ delegate symbolic/numeric computation and typesetting of documents to computers.

- ▷ (sometimes) delegate proof checking and search for trivial proofs to computers

Overlooked Opportunity: management of existing mathematical knowledge

- ▷ ▹ cataloguing, retrieval, refactoring, plausibilization, change propagation and in some cases even application do not require (human) insights and intuition
  - ▹ can even be automated in the near future given suitable representation formats and algorithms.

Math. Knowledge Management (MKM): is the discipline that studies this.

⊳ Application: Scaling Math beyond the One-Brain-Barrier

The One-Brain-Barrier

- ▷ **Observation 2.1.5** *More than 10^5 math articles published annually in Math.*
- ▷ **Observation 2.1.6** *The libraries of Mizar, Coq, Isabelle,... have $\sim 10^5$ statements+proofs each. (but are mutually incompatible)*
- ▷ Consequence: humans lack overview over – let alone working knowledge in – all of math/formalizations. (Leonardo da Vinci was said to be the last who had)
- ▷ Dire Consequences: duplication of work and missed opportunities for the application of mathematical/formal results.
- ▷ Problem: Math Information systems like `arXiv.org`, Zentralblatt Math, MathSciNet, etc. do not help (only make documents available)
- ▷ Fundamenal Problem: the One-Brain Barrier (OBB)
  - ▹ To become productive, math must pass through a brain
  - ▹ Human brains have limited capacity (compared to knowledge available online)
- ▷ Idea: enlist computers (large is what they are good at)
- ▷ Prerequisite: make math knowledge machine-actionable & foundation-independent (use MKM)

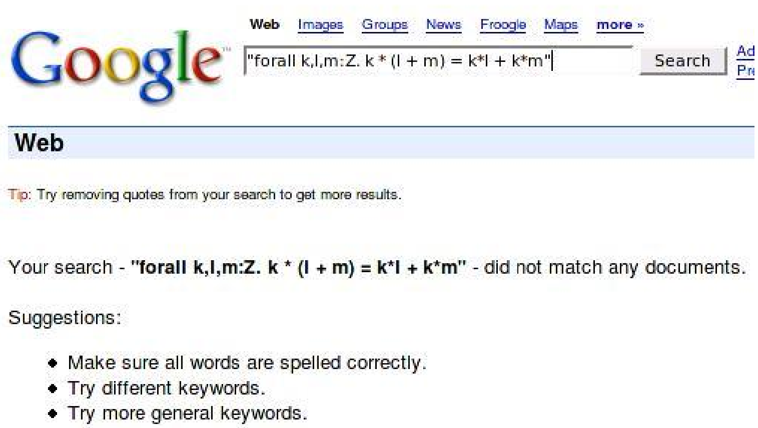
All of that is very abstract, high-level and idealistic, ... Let us look at an example, where we can see computer support for one of the postulated horizontal/MKM tasks in action.

2.2 Mathematical Formula Search

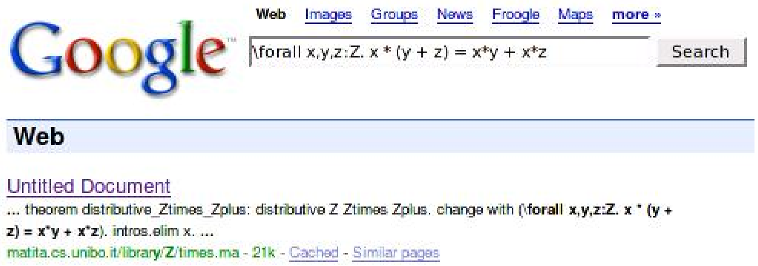
More Mathematics on the Web

- ▷ The Connexions project (http://cnx.org)
- ▷ Wolfram Inc. (http://functions.wolfram.com)
- ▷ Eric Weisstein's MathWorld (http://mathworld.wolfram.com)
- ▷ Digital Library of Mathematical Functions (http://dlmf.nist.gov)
- ▷ Cornell ePrint arXiv (http://www.arxiv.org)
- ▷ Zentralblatt Math (http://www.zentralblatt-math.org)
- ▷ . . . Engineering Company Intranets, . . .
- ▷ Question: How will we find content that is relevant to our needs
- ▷ Idea: try Google (like we always do)
- ▷ Scenario: Try finding the distributivity property for $\mathbb{Z}$ ($\forall k, l, m \in \mathbb{Z}. k \cdot (l + m) = (k \cdot l) + (k + m)$)

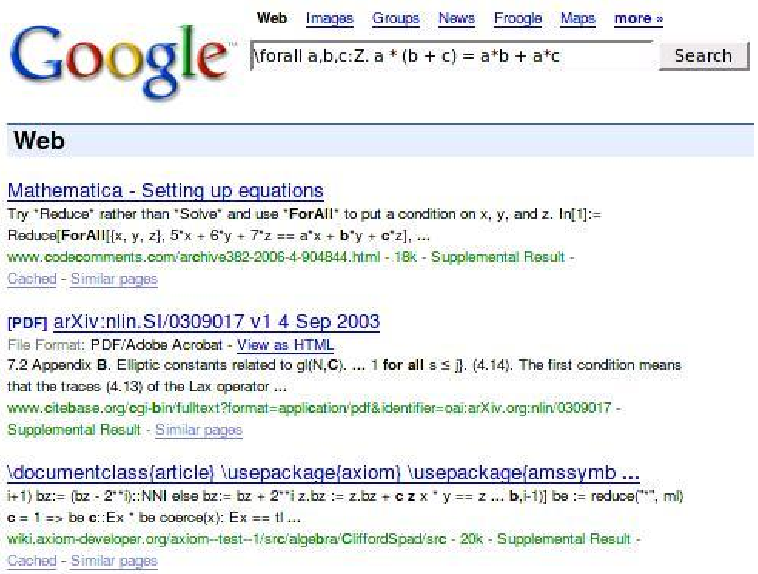
Searching for Distributivity



Searching for Distributivity

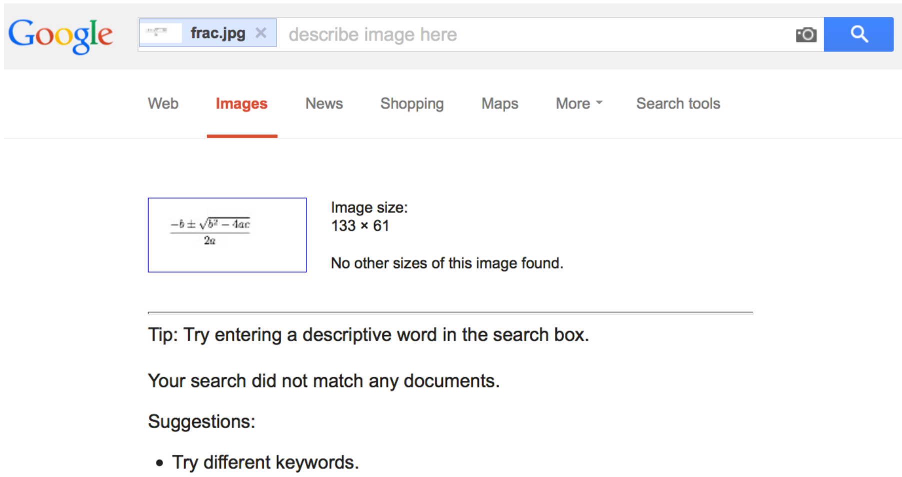


Searching for Distributivity



Does Image Search help?

▷ Math formulae are visual objects, after all (let's try it)



Of course Google cannot work out of the box

- Formulae are not words:
  - a, b, c, k, l, m, x, y, and z are (bound) variables. (do not behave like words/symbols)
  - where are the word boundaries for "bag-of-words" methods?
- Formulae are not images either: They have internal (recursive) structure and compositional meaning
- Idea: Need a special treatment for formulae (translate into "special words")
  Indeed this is done ([MY03; MM06; LM06; MG11])
  . . . and works surprisingly well (using e.g. Lucene as an indexing engine)
- Idea: Use database techniques (extract metadata and index it)
  Indeed this is done for the `Coq`/`HELM` corpus ([Asp+06])
- Our Idea: Use Automated Reasoning Techniques (free term indexing from theorem prover jails)
- Demo: MathWebSearch on Zentralblatt Math, the arXiv Data Set

A running example: The Power of a Signal

- An engineer wants to compute the power of a given signal $s(t)$
- She remembers that it involves integrating the square of s.
- Problem: But how to compute the necessary integrals

- Idea: call up `MathWebSearch` with $\int_?^? s^2(t)dt$.
- `MathWebSearch` finds a document about Parseval's Theorem and $\frac{1}{T}\int_0^T s^2(t)dt = \Sigma_{k=-\infty}^{\infty}|c_k|^2$ where c_k are the Fourier coefficients of $s(t)$.

Some other Problems (Why do we need more?)

- Substitution Instances: search for $x^2 + y^2 = z^2$, find $3^2 + 4^2 = 5^2$
- Homonymy: $\binom{n}{k}$, ${}_nC^k$, C_k^n, C_n^k, and ${}_k\mathcal{J}^n$ all mean the same thing (binomial coeff.)
- Solution: use content-based representations (MathML, $OpenMath$)
- Mathematical Equivalence: e.g. $\int f(x)dx$ means the same as $\int f(y)dy$ (α-equivalence)
- Solution: build equivalence (e.g. α or ACI) into the search engine (or normalize first [Normann'06])
- Subterms: Retrieve formulae by specifying some sub-formulae
- Solution: record locations of all sub-formulae as well

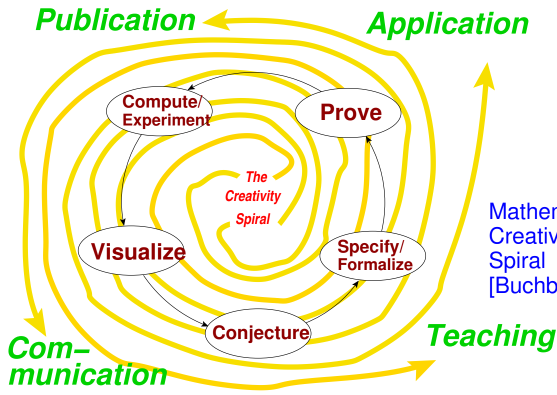
`MathWebSearch`: Search Math. Formulae on the Web

- Idea 1: Crawl the Web for math. formulae (in $OpenMath$ or CMathML)
- Idea 2: Math. formulae can be represented as first order terms (see below)
- Idea 3: Index them in a substitution tree index (for efficient retrieval)
- Problem: Find a query language that is intuitive to learn
- Idea 4: Reuse the XML syntax of $OpenMath$ and CMathML, add variables

2.3 The Mathematical Knowledge Space

The way we do math will change dramatically

- **Definition 2.3.1 (Doing Math)** Buchberger's Math creativity spiral



Mathematical Creativity Spiral [Buchberger 1995]

▷ Every step will be supported by mathematical software systems

▷ Towards an infrastructure for web-based mathematics!

Mathematical Literacy

▷ Note: the form and extent of knowledge representation for the components of "doing math" vary greatly. (e.g. publication vs. proving)

▷ **Observation 2.3.2 (Primitive Cognitive Actions)**
To "do mathematics", we need to

- *extract the relevant structures,*
- *reconcile them with the context of our existing knowledge*
- *recognize parts as already known*
- *identify parts that are new to us.*

During these processes mathematicians (are trained to)

- *abstract from syntactic differences, and*
- *employ interpretations via non-trivial, but meaning-preserving mappings*

▷ **Definition 2.3.3** We call the skillset that identifies mathematical training mathematical literacy (cf. Observation 2.3.2)

Introduction: Framing as a Mathematical Practice

▷ Understanding Mathematical Practices:

- To understand Math, we must understand what mathematicians do!
- The value of a math education is more in the skills than in the knowledge.
- Have been interested in this for a while (see [KK06])

▷ Framing: Understand new objects in terms of already understood structures. Make creative use of this perspective in problem solving.

▷ **Example 2.3.4** Understand point sets in 3-space as zeroes of polynomials. Derive insights by studying the algebraic properties of polynomials.

▷ **Definition 2.3.5** We are framing the point sets as algebraic varieties (sets of zeroes of polynomials).

▷ **Example 2.3.6 (Lie group)** Equipping a differentiable manifold with a (differentiable) group operation

▷ **Example 2.3.7 (Stone's representation theorem)** Interpreting a Boolean algebra as a field of sets.

▷ Claim: Framing is valuable, since it transports insights between fields.

▷ Claim: Many famous theorems earn their recognition *because* they establish profitable framings.

 23

2.4 Modular Representation of mathematical Knowledge

Modular Representation of Math (Theory Graph)

▷ Idea: Follow mathematical practice of generalizing and framing

- ▷ framing: If we can view an object a as an instance of concept B, we can inherit all of B properties (almost for free.)
- ▷ state all assertions about properties as general as possible (to maximize inheritance)
- ▷ examples and applications are just special framings.

▷ Modern expositions of Mathematics follow this rule (radically e.g. in Bourbaki)

▷ formalized in the theory graph paradigm (little/tiny theory doctrine)

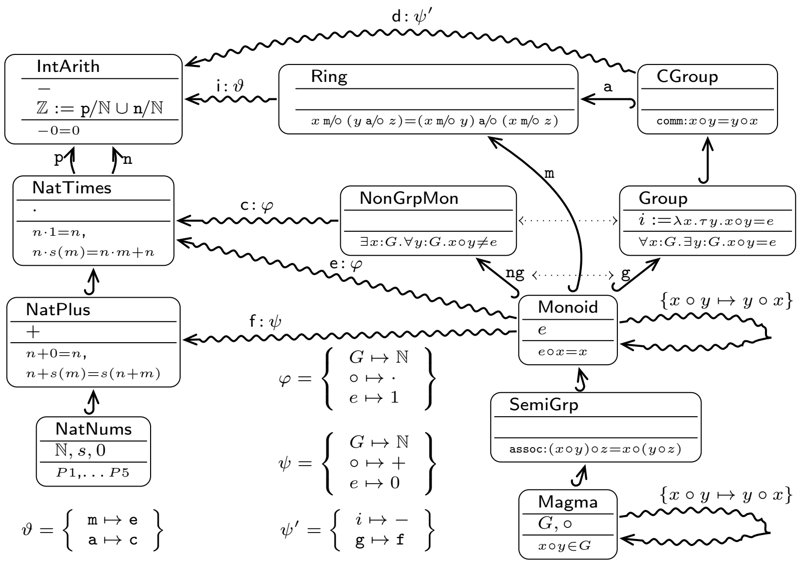
- ▷ theories as collections of symbol declarations and axioms (model assumptions)
- ▷ theory morphisms as mappings that translate axioms into theorems

▷ **Example 2.4.1 (MMT: Modular Mathematical Theories)** MMT is a foundation-indepent theory graph formalism with advanced theory morphisms.

Problem: With a proliferation of abstract (tiny) theories readability and accessibility suffers (one reason why the Bourbaki books fell out of favor)

 24

▷ Modular Representation of Math (MMT Example)

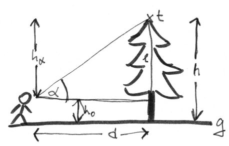


2.5 Application: Serious Games

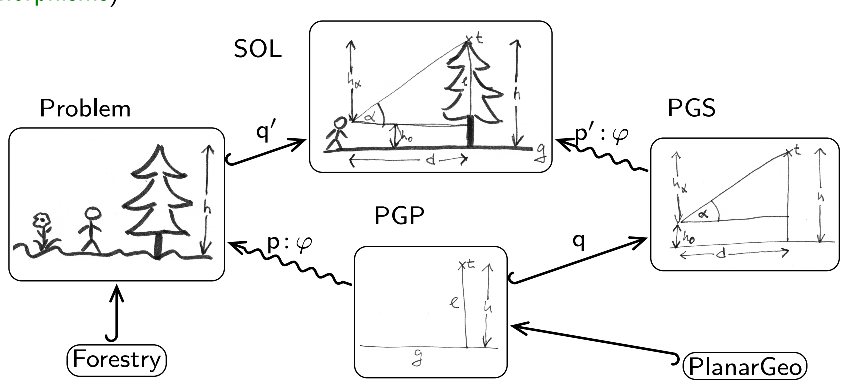
Framing for Problem Solving (The FrameIT Method)

▷ **Example 2.5.1 (Problem 0.8.15)**

How can you measure the height of a tree you cannot climb, when you only have a protractor and a tape measure at hand.

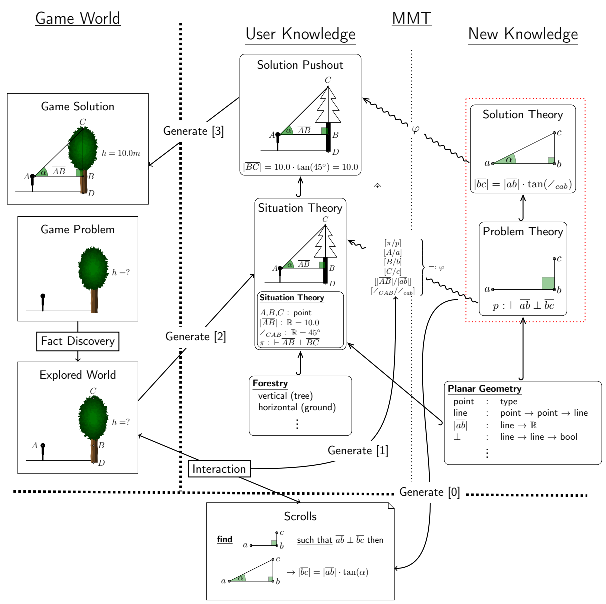


▷ Framing: view the problem as one that is already understood (using theory morphisms)



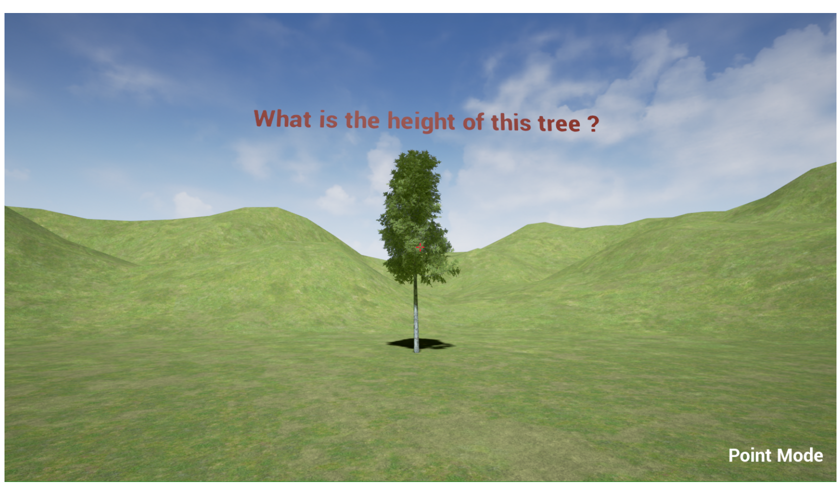
▷ squiggly (framing) morphisms guaranteed by metatheory of theories!

Example Learning Object Graph



FrameIT Method: Problem

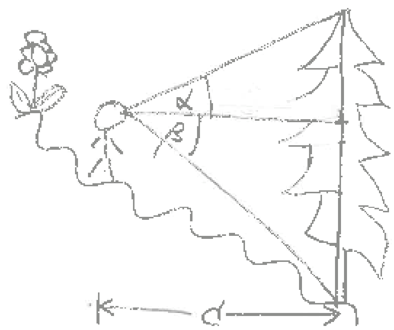
▷ Problem Representation in the game world (what the student should see)



▷ Student can interact with the environment via gadgets so solve problems

▷ "Scrolls" of mathematical knowledge give hints.

Combining Problem/Solution Pairs



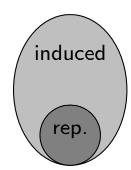
- ▷ We can use the same mechanism for combining P/S pairs
- ▷ create more complex P/S pairs (e.g. for trees on slopes)

Another whole set of applications and game behaviors can come from the fact that LOGraphs give ways to combine problem/solution pairs to novel ones. Consider for instance the diagram on the right, where we can measure the height of a tree of a slope. It can be constructed by combining the theory SOL with a copy of SOL along a second morphism the inverts h to $-h$ (for the lower triangle with angle β) and identifies the base lines (the two occurrences of h_0 cancel out). Mastering the combination of problem/solution pairs further enhances the problem solving repertoire of the player.

2.6 Search in the Mathematical Knowledge Space

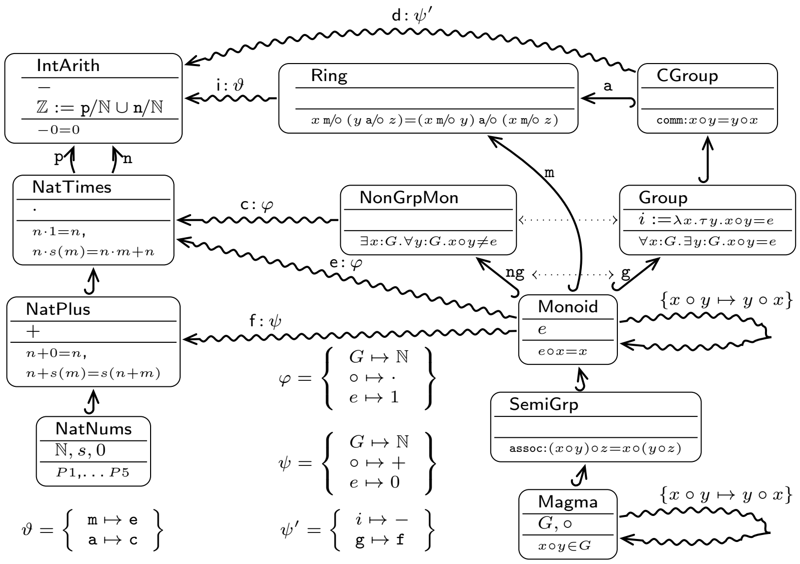
The Mathematical Knowledge Space

- ▷ **Observation 2.6.1** *The value of framing is that it induces new knowledge*
- ▷ **Definition 2.6.2** The mathematical knowledge space MKS is the structured space of represented and induced knowledge, mathematically literate have access to.



- ▷ Idea: make math systems mathematically literate by supporting the MKS
- ▷ In this talk: I will cover three aspects
  - ▷ an approach for representing framing and the MKS (OMDoc/MMT)
  - ▷ search modulo framing (MKS-literate search)
  - ▷ a system for archiving the MKS (MathHub.info)
- ▷ Told from the Perspective of: searching the MKS

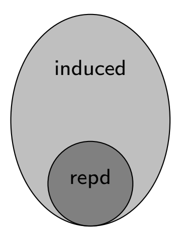
Modular Representation of Math (MMT Example)



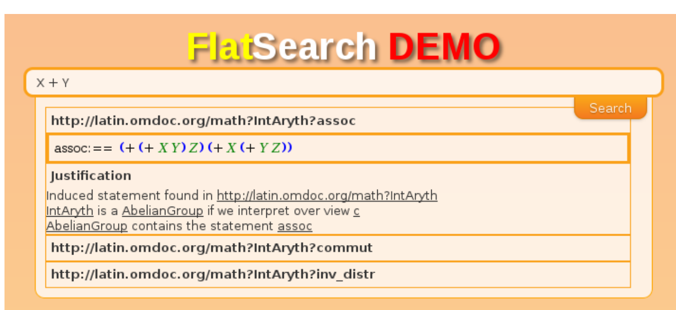
♭search on the LATIN Logic Atlas

▷ Flattening the LATIN Atlas (once):

type	modular	flat	factor
declarations	2310	58847	25.4
library size	23.9 MB	1.8 GB	14.8
math sub-library	2.3 MB	79 MB	34.3
`MathWebSearch` harvests	25.2 MB	539.0 MB	21.3



▷ simple ♭search frontend at http://cds.omdoc.org:8181/search.html



Overview: KWARC Research and Projects

Applications: eMath 3.0, Active Documents, Semantic Spreadsheets, Semantic CAD/CAM, Change Mangagement, Global Digital Math Library, Math Search Systems, SMGloM: Semantic Multilingual Math Glossary, Serious Games, ...		
Foundations of Math: ▷ MathML, *OpenMath* ▷ advanced Type Theories ▷ MMT: Meta Meta Theory ▷ Logic Morphisms/Atlas ▷ Theorem Prover/CAS Interoperability	**KM & Interaction**: ▷ Semantic Interpretation (aka. Framing) ▷ math-literate interaction ▷ MathHub: math archives & active docs ▷ Semantic Alliance: embedded semantic services	**Semantization**: ▷ LaTeXML: LaTeX → XML ▷ sTeX: Semantic LaTeX ▷ invasive editors ▷ Context-Aware IDEs ▷ Mathematical Corpora ▷ Linguistics of Math
Foundations: Computational Logic, Web Technologies, *OMDoc*/MMT		

Take-Home Message

- ▷ Overall Goal: *Overcoming the "One-Brain-Barrier" in Mathematics* (by knowledge-based systems)
- ▷ Means: Mathematical Literacy by Knowledge Representation and Processing in theory graphs in Theoriegraphen. (Framing as mathematical practice)

Chapter 3

What is (Computational) Logic

What is (Computational) Logic?

- ▷ The field of logic studies representation languages, inference systems, and their relation to the world.
- ▷ It dates back and has its roots in Greek philosophy (Aristotle et al.)
- ▷ Logical calculi capture an important aspect of human thought, and make it amenable to investigation with mathematical rigour, e.g. in
  - ▷ foundation of mathematics (Hilbert, Russell and Whitehead)
  - ▷ foundations of syntax and semantics of language(Creswell, Montague, ...)
- ▷ Logics have many practical applications
  - ▷ logic/declarative programming (the third programming paradigm)
  - ▷ program verification: specify conditions in logic, prove program correctness
  - ▷ program synthesis: prove existence of answers constructively, extract program from proof
  - ▷ proof-carrying code: compiler proves safety conditions, user verifies before running.
  - ▷ deductive databases: facts + rules (get more out than you put in)
  - ▷ semantic web: the Web as a deductive database
- ▷ Computational Logic is the study of logic from a computational, proof-theoretic perspective. (model theory is mostly comprised under "mathematical logic".)

What is Logic?

- ▷ Logic $\widehat{=}$ formal languages, inference and their relation with the world
  - ▷ Formal language $\mathcal{FL}$: set of formulae ($2 + 3/7$, $\forall x.x + y = y + x$)
  - ▷ Formula: sequence/tree of symbols ($x, y, f, g, p, 1, \pi, \in, \neg, \wedge \forall, \exists$)

- Model: things we understand (e.g. number theory)
- Interpretation: maps formulae into models ($[\![\text{three plus five}]\!] = 8$)
- Validity: $\mathcal{M} \models \mathbf{A}$, iff $[\![\mathbf{A}]\!]^{\mathcal{M}} = \mathsf{T}$ (five greater three is valid)
- Entailment: $\mathbf{A} \models \mathbf{B}$, iff $\mathcal{M} \models \mathbf{B}$ for all $\mathcal{M} \models \mathbf{A}$. (generalize to $\mathcal{H} \models \mathbf{A}$)
- Inference rules to transform (sets of) formulae ($\mathbf{A}, \mathbf{A} \Rightarrow \mathbf{B} \vdash \mathbf{B}$)
- Syntax: formulae, inference (just a bunch of symbols)
- Semantics: models, interpr., validity, entailment (math. structures)

Important Question: relation between syntax and semantics?

So logic is the study of formal representations of objects in the real world, and the formal statements that are true about them. The insistence on a *formal language* for representation is actually something that simplifies life for us. Formal languages are something that is actually easier to understand than e.g. natural languages. For instance it is usually decidable, whether a string is a member of a formal language. For natural language this is much more difficult: there is still no program that can reliably say whether a sentence is a grammatical sentence of the English language.

We have already discussed the meaning mappings (under the monicker "semantics"). Meaning mappings can be used in two ways, they can be used to understand a formal language, when we use a mapping into "something we already understand", or they are the mapping that legitimize a representation in a formal language. We understand a formula (a member of a formal language) $\mathbf{A}$ to be a representation of an object $\mathcal{O}$, iff $[\![\mathbf{A}]\!] = \mathcal{O}$.

However, the game of representation only becomes really interesting, if we can do something with the representations. For this, we give ourselves a set of syntactic rules of how to manipulate the formulae to reach new representations or facts about the world.

Consider, for instance, the case of calculating with numbers, a task that has changed from a difficult job for highly paid specialists in Roman times to a task that is now feasible for young children. What is the cause of this dramatic change? Of course the formalized reasoning procedures for arithmetic that we use nowadays. These *calculi* consist of a set of rules that can be followed purely syntactically, but nevertheless manipulate arithmetic expressions in a correct and fruitful way. An essential prerequisite for syntactic manipulation is that the objects are given in a formal language suitable for the problem. For example, the introduction of the decimal system has been instrumental to the simplification of arithmetic mentioned above. When the arithmetical calculi were sufficiently well-understood and in principle a mechanical procedure, and when the art of clock-making was mature enough to design and build mechanical devices of an appropriate kind, the invention of calculating machines for arithmetic by Wilhelm Schickard (1623), Blaise Pascal (1642), and Gottfried Wilhelm Leibniz (1671) was only a natural consequence.

We will see that it is not only possible to calculate with numbers, but also with representations of statements about the world (propositions). For this, we will use an extremely simple example; a fragment of propositional logic (we restrict ourselves to only one logical connective) and a small calculus that gives us a set of rules how to manipulate formulae.

3.1 A History of Ideas in Logic

Before starting with the discussion on particular logics and inference systems, we put things into perspective by previewing ideas in logic from a historical perspective. Even though the presentation (in particular syntax and semantics) may have changed over time, the underlying ideas are still pertinent in today's formal systems.

Many of the source texts of the ideas summarized in this Section can be found in [Hei67].

▷ History of Ideas (abbreviated): Propositional Logic

- ▷ General Logic ([ancient Greece, e.g. Aristotle])
  - + conceptual separation of syntax and semantics
  - + system of inference rules ("Syllogisms")
  - – no formal language, no formal semantics
- ▷ Propositional Logic [Boole $\sim$ 1850]
  - + functional structure of formal language (propositions + connectives)
  - + mathematical semantics ($\leadsto$ Boolean Algebra)
  - – abstraction from internal structure of propositions

History of Ideas (continued): Predicate Logic

- ▷ Frege's "Begriffsschrift" [Fre79]
  - + functional structure of formal language (terms, atomic formulae, connectives, quantifiers)
  - – weird graphical syntax, no mathematical semantics
  - – paradoxes e.g. Russell's Paradox [R. 1901] (the set of sets that do not contain themselves)
- ▷ modern form of predicate logic [Peano $\sim$ 1889]
  - + modern notation for predicate logic ($\vee, \wedge, \Rightarrow, \forall, \exists$)

History of Ideas (continued): First-Order Predicate Logic

- ▷ Types ([Russell 1908])
  - – restriction to well-types expression
  - + paradoxes cannot be written in the system
  - + Principia Mathematica ([Whitehead, Russell 1910])
- ▷ Identification of first-order Logic ([Skolem, Herbrand, Gödel $\sim$ 1920 – '30])
  - – quantification only over individual variables (cannot write down induction principle)
  - + correct, complete calculi, semi-decidable
  - + set-theoretic semantics ([Tarski 1936])

History of Ideas (continued): Foundations of Mathematics

- Hilbert's Program: find logical system and calculus, ([Hilbert ∼ 1930])
  - that formalizes all of mathematics
  - that admits sound and complete calculi
  - whose consistence is provable in the system itself
- Hilbert's Program is impossible! ([Gödel 1931])

  Let $\mathcal{L}$ be a logical system that formalizes arithmetics ($\langle NaturalNumbers, +, * \rangle$),
  - then $\mathcal{L}$ is incomplete
  - then the consistence of $\mathcal{L}$ cannot be proven in $\mathcal{L}$.

History of Ideas (continued): λ-calculus, set theory

- Simply typed λ-calculus ([Church 1940])
  - \+ simplifies Russel's types, λ-operator for functions
  - \+ comprehension as β-equality (can be mechanized)
  - \+ simple type-driven semantics (standard semantics $\leadsto$ incompleteness)
- Axiomatic set theory
  - +– type-less representation (all objects are sets)
  - \+ first-order logic with axioms
  - \+ restricted set comprehension (no set of sets)
  - – functions and relations are derived objects

Part I

Excursions

As this course is predominantly about modeling mathematical knowledge and not about the theoretical aspects of the logics we use for that themselves, we give the discussion about these as a "suggested readings" section part here.

Bibliography

[Asp+06] Andrea Asperti et al. "A Content Based Mathematical Search Engine: Whelp". In: *Types for Proofs and Programs, International Workshop, TYPES 2004, revised selected papers*. Ed. by Jean-Christophe Filliâtre, Christine Paulin-Mohring, and Benjamin Werner. LNCS 3839. Springer Verlag, 2006, pp. 17–32.

[Fre79] Gottlob Frege. *Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. 1879.

[Hei67] Jean van Heijenoort. *From Frege to Gödel: a source book in mathematical logic 1879-1931*. 3rd printing, 1997. Source books in the history of the sciences series. Cambridge, MA: Harvard Univ. Press, 1967. ISBN: 0-674-32450-1.

[Jin10] Arif Jinha. "Article 50 million: an estimate of the number of scholarly articles in existence". In: *Learned Publishing* 23.3 (2010), pp. 258–263. DOI: 10.1087/20100308.

[KK06] Andrea Kohlhase and Michael Kohlhase. "Communities of Practice in MKM: An Extensional Model". In: *Mathematical Knowledge Management (MKM)*. Ed. by Jon Borwein and William M. Farmer. LNAI 4108. Springer Verlag, 2006, pp. 179–193. URL: http://kwarc.info/kohlhase/papers/mkm06cp.pdf.

[Koh08] Michael Kohlhase. "Using LaTeX as a Semantic Markup Format". In: *Mathematics in Computer Science* 2.2 (2008), pp. 279–304. URL: https://kwarc.info/kohlhase/papers/mcs08-stex.pdf.

[Koh17] Michael Kohlhase. *sTeX: Semantic Markup in TeX/LaTeX*. Tech. rep. Comprehensive TeX Archive Network (CTAN), 2017. URL: http://www.ctan.org/get/macros/latex/contrib/stex/sty/stex.pdf.

[LI10] Peder Olesen Larsen and Markus von Ins. "The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index". In: *Scientometrics* 84.3 (2010), pp. 575–603. DOI: 10.1007/s11192-010-0202-z.

[LM06] Paul Libbrecht and Erica Melis. "Methods for Access and Retrieval of Mathematical Content in ActiveMath". In: *Proceedings of ICMS-2006*. Ed. by N. Takayama and A. Iglesias. LNAI 4151. http://www.activemath.org/publications/Libbrecht-Melis-Access-and-Retrieval-ActiveMath-ICMS-2006.pdf. Springer Verlag, 2006, pp. 331–342. URL: http://www.activemath.org/publications/Libbrecht-Melis-Access-and-Retrieval-ActiveMath-ICMS-2006.pdf.

[MG11] Jozef Misutka and Leo Galambos. "System Description: EgoMath2 As a Tool for Mathematical Searching on Wikipedia.org". In: *Calculemus/MKM*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 307–309. ISBN: 978-3-642-22672-4.

[MM06] Rajesh Munavalli and Robert Miner. "MathFind: a math-aware search engine". In: *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. Seattle, Washington, USA: ACM Press, 2006, pp. 735–735. ISBN: 1-59593-369-7. DOI: http://doi.acm.org/10.1145/1148170.1148348.

[MY03] Bruce R. Miller and Abdou Youssef. "Technical Aspects of the Digital Library of Mathematical Functions". In: *Annals of Mathematics and Artificial Intelligence* 38.1-3 (2003), pp. 121–136. URL: citeseer.ist.psu.edu/599441.html.

[OMT] Michael Kohlhase and Dennis Müller. *OMDoc/MMT Tutorial for Mathematicians*. URL: https://gl.mathhub.info/Tutorials/Mathematicians/blob/master/tutorial/mmt-math-tutorial.pdf (visited on 10/07/2017).

Index