Knowledge Representation for Mathematics & Technology

Prof. Dr. Michael Kohlhase & PD. Dr. Florian Rabe

Professur für Wissensrepräsentation und -verarbeitung
Informatik, FAU Erlangen-Nürnberg
Michael.Kohlhase,Florian.Rabe@FAU.de

2023-04-25

KRMT: Course Concept

- **This course will teach you:**
  - **Theory**: foundations of mathematics, syntax/semantics/proof theory of multiple logics, meta-logical frameworks.
  - **Practice**: modular formalizations of math in theory graphs, development of logics, inference systems and mechanizations.
  - Anything others can do you can then do meta!

KRMT: Course Concept

- **This course will teach you:**
  - **Theory**: foundations of mathematics, syntax/semantics/proof theory of multiple logics, meta-logical frameworks.
  - **Practice**: modular formalizations of math in theory graphs, development of logics, inference systems and mechanizations.
  - Anything others can do you can then do meta!
- **Teaching Concept:** Small course with lectures/labs
  - **Theory**: lectures with lots of discussions (∼ tuesdays)
  - **Practice**: jointly formalizing math/logics (∼ wednesdays)

KRMT: Course Concept

- **This course will teach you:**
  - **Theory**: foundations of mathematics, syntax/semantics/proof theory of multiple logics, meta-logical frameworks.
  - **Practice**: modular formalizations of math in theory graphs, development of logics, inference systems and mechanizations.
  - Anything others can do you can then do meta!
- **Teaching Concept:** Small course with lectures/labs
  - **Theory**: lectures with lots of discussions ($\sim$ tuesdays)
  - **Practice**: jointly formalizing math/logics ($\sim$ wednesdays)
- **Course Goal:**
  Recruiting and grooming junior researchers for KWARC
  - Come do research with us, we have good supervision and fascinating topics!

0.1 Administrativa

Prerequisites for KRMT

- **Content Prerequisites:** the mandatory courses in CS@FAU; Sem 1-4, in particular:
  - course “Grundlagen der Logik in der Informatik” (GLOIN)
  - CS Math courses “Mathematik C1-4” (IngMath1-4) (our “domain”)
  - algorithms and data structures
  - AI-1 (“Artificial Intelligence I”) (nice-to-have only)

- You can do this course if you want! (We will help you)

Prerequisites for KRMT

- **Content Prerequisites:** the mandatory courses in CS@FAU; Sem 1-4, in particular:
  - course “Grundlagen der Logik in der Informatik” (GLOIN)
  - CS Math courses “Mathematik C1-4” (IngMath1-4) (our “domain”)
  - algorithms and data structures
  - AI-1 (“Artificial Intelligence I”) (nice-to-have only)
- **Intuition:** (take them with a kilo of salt)
  - This is what I assume you know! (I have to assume something)
  - In many cases, the dependency of KRMT on these is partial and “in spirit”.
  - If you have not taken these (or do not remember),
    - read up on them as needed! (preferred, do it in a group)
    - We can cover them in class (if there are more of you)
- You can do this course if you want! (We will help you)

Prerequisites for KRMT

- **Content Prerequisites:** the mandatory courses in CS@FAU; Sem 1-4, in particular:
  - course “Grundlagen der Logik in der Informatik” (GLOIN)
  - CS Math courses “Mathematik C1-4” (IngMath1-4) (our “domain”)
  - algorithms and data structures
  - AI-1 (“Artificial Intelligence I”) (nice-to-have only)
- **Intuition:** (take them with a kilo of salt)
  - This is what I assume you know! (I have to assume something)
  - In many cases, the dependency of KRMT on these is partial and “in spirit”.
  - If you have not taken these (or do not remember),
    - read up on them as needed! (preferred, do it in a group)
    - We can cover them in class (if there are more of you)
- **The real Prerequisite:** Motivation, Interest, Curiosity, hard work. (KRMT is non-trivial)
- You can do this course if you want! (We will help you)

KRMT Lab (Dogfooding our own Techniques)

- **Underlying Problem:** There are about 20 deep results/insights/tricks necessary to understand KRMT.
- **The Good News:** These are sufficient too, if you can apply them (non-trivial)
- **Consequence:** KRMT may be the course with the highest "pain-per-letter ratio" (but it is wonderful when the pain goes away)

KRMT Lab (Dogfooding our own Techniques)

- **Underlying Problem:** There are about 20 deep results/insights/tricks necessary to understand KRMT.
- **The Good News:** These are sufficient too, if you can apply them (non-trivial)
- **Consequence:** KRMT may be the course with the highest "pain-per-letter ratio" (but it is wonderful when the pain goes away)
- **General Plan:** We use the Wednesday slot to get our hands dirty with actual MMT formalizations.
- **Goal:** Reinforce what was taught on Tuesdays and have some fun.
- **How this works:** we jointly develop key formalizations in class
  - we discuss the pertinent issues, you dictate, we test in the system.
  - what is left over becomes homework (the routine parts)
  - we discuss problems, ... on the KRMT chat (details later)
- **Caveat:** Only by practical involvement will you be able to understand the difficult theoretical issues/ideas! (so come and participate)

Homeworks

- **Goal:** Homework assignments/problems reinforce what was taught in Lectures/Labs
- **Homeworks** will be small individual formalization tasks (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.
  - We can be flexible about deadlines (but deadlines help you)

Grades (Academic Assessment)

- **What we used so far:** two parts (Portfolio Assessment)
  - 20-30 min oral exam at the end of the semester (50%)
  - results of the KRMT lab (50%)

  This will not work with 50+ students, need to see how the course develops!
- **How about this:** three parts (Portfolio Assessment)
  - 60 min written exam early October? (70%)
  - results of the KRMT lab (30%)
  - bonus project after the semester (10% bonus)
- If you have suggestions, I will probably be happy with that as well.
- Let's finalize this next week.

Textbook, Handouts and Information, Forums, Chat

- **(No) Textbook:** there is none!
  - Course notes will be posted at `http://kwarc.info/teaching/KRMT`
  - We mostly prepare/update them as we go along (semantically preloaded ⇝ research resource)
  - Please e-mail us any errors/shortcomings you notice. (improve for the group)
- The KRMT lab generally follows the MMT tutorial at `https://gl.mathhub.info/Tutorials/Mathematicians/blob/master/tutorial/mmt-math-tutorial.pdf`
- Announcements will be posted on the course forum
  - `https://www.studon.fau.de/frm5126852.html`
- Check the forum frequently for (adopt/use it, this is for you!)
  - announcements, homeworks, questions
  - discussion among your fellow students
- We have to choose a chat venue (Matrix or StudOn)

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.

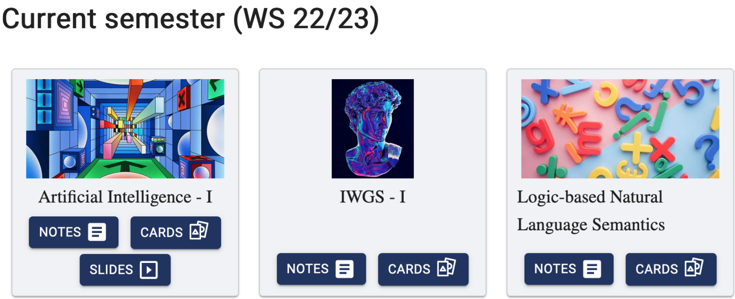
  The only advantage of I over B is that books/papers do not answer questions
- Approach S: come to the lectures and sleep does not work!
- The closer you get to research, the more we need to discuss!

Experiment: Learning Support with KWARC Technologies

- **My research area:** Deep representation formats for (mathematical) knowledge
- **One Application:** Learning support systems (represent knowledge to transport it)
- **Experiment:** Start with this course (Drink my own medicine)
  1. Re-Represent the slide materials in OMDoc (Open Mathematical Documents)
  2. Feed it into the ALeA system (`http://courses.voll-ki.fau.de`)
  3. Try it on you all (to get feedback from you)
- Tasks (I cannot pay you for this)
  - 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)

VoLL-KI Portal at https://courses.voll-ki.fau.de

- **Idea:** Provide HTML versions of the slides/notes and embed learning support services into them. (for pre/postparation of lectures)



Current semester (WS 22/23)

Artificial Intelligence - I

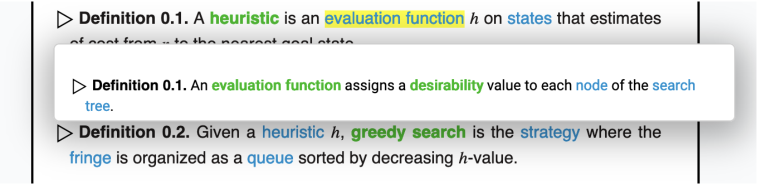
IWGS - I

Logic-based Natural Language Semantics

- **Definition 1.1.** Call a document active, iff it is interactive and adapts to specific information needs of the readers. (course notes on steroids)

VoLL-KI Portal at https://courses.voll-ki.fau.de

- **Idea:** Provide HTML versions of the slides/notes and embed learning support services into them. (for pre/postparation of lectures)
- **Definition 1.4.** Call a document active, iff it is interactive and adapts to specific information needs of the readers. (course notes on steroids)
- **Example 1.5 (Definition on Hover).** When we hover on a (cyan) term reference, hovering shows us the definition. (even works recursively)



▷ **Definition 0.1.** A heuristic is an evaluation function h on states that estimates of cost from n to the nearest goal state

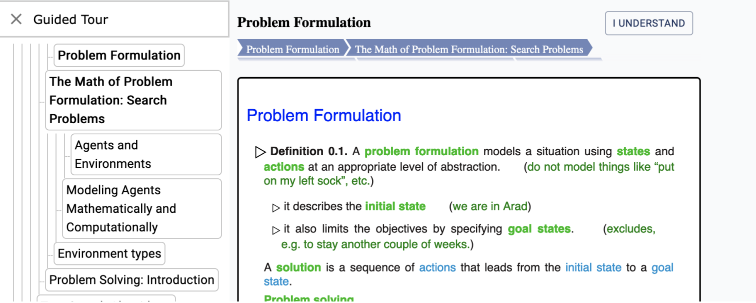
▷ **Definition 0.1.** An evaluation function assigns a desirability value to each node of the search tree.

▷ **Definition 0.2.** Given a heuristic h, greedy search is the strategy where the fringe is organized as a queue sorted by decreasing h-value.

When we click on the hover popup, we get even more information!

VoLL-KI Portal at https://courses.voll-ki.fau.de

- **Idea:** Provide HTML versions of the slides/notes and embed learning support services into them. (for pre/postparation of lectures)
- **Definition 1.7.** Call a document active, iff it is interactive and adapts to specific information needs of the readers. (course notes on steroids)
- **Example 1.8 (Definition on Hover).** When we hover on a (cyan) term reference, hovering shows us the definition. (even works recursively)
  When we click on the hover popup, we get even more information!
- **Example 1.9 (Guided Tour).** A guided tour for a concept *c* assembles definitions/etc. into a self-contained mini-course culminating at *c*.



× Guided Tour
Problem Formulation
The Math of Problem Formulation: Search Problems
Agents and Environments
Modeling Agents Mathematically and Computationally
Environment types
Problem Solving: Introduction
I UNDERSTAND
Problem Formulation
The Math of Problem Formulation: Search Problems

Problem Formulation

Problem Formulation

▷ **Definition 0.1.** A **problem formulation** models a situation using **states** and **actions** at an appropriate level of abstraction. (do not model things like "put on my left sock", etc.)

- ▷ it describes the **initial state** (we are in Arad)
- ▷ it also limits the objectives by specifying **goal states**. (excludes, e.g. to stay another couple of weeks.)

A **solution** is a sequence of actions that leads from the initial state to a goal state.

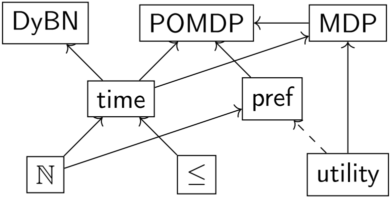
Problem solving

VoLL-KI Portal at https://courses.voll-ki.fau.de

- **Idea:** Provide HTML versions of the slides/notes and embed learning support services into them. (for pre/postparation of lectures)
- **Definition 1.10.** Call a document active, iff it is interactive and adapts to specific information needs of the readers. (course notes on steroids)
- **Example 1.11 (Definition on Hover).** When we hover on a (cyan) term reference, hovering shows us the definition. (even works recursively) When we click on the hover popup, we get even more information!
- **Example 1.12 (Guided Tour).** A guided tour for a concept *c* assembles definitions/etc. into a self-contained mini-course culminating at *c*.
- **Status:** The ALeA system is deployed at FAU for over 1000 students taking six courses
  - (some) students use the system actively (our logs tell us)
  - reviews are mostly positive/enthusiastic (error reports pour in)

ALeA$\widehat{=}$ Data-Driven & AI-enabled Learning Assistance

- **Ingredient 1:** Domain model $\widehat{=}$ knowledge/theory graph

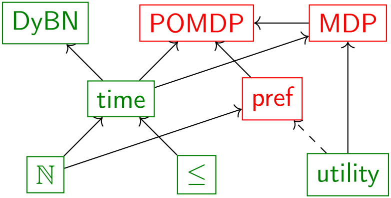


A theory graph provides (modular representation of the domain)

- symbols with URIs for all concepts, objects, and relations
- definitions, notations, and verbalizations for all symbols
- “object-oriented inheritance” and views between theories.

ALeA$\widehat{=}$ Data-Driven & AI-enabled Learning Assistance

- **Ingredient 1:** Domain model $\widehat{=}$ knowledge/theory graph
- **Ingredient 2:** Learner model $\widehat{=}$ adding competency estimations

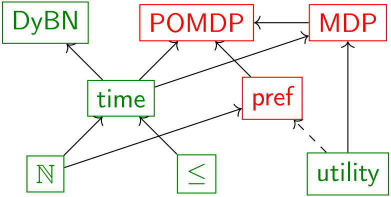


The learner model is a function from learner IDs × symbol URIs to competency values

- competency comes in six cognitive dimensions: `remember`, `understand`, `analyze`, `evaluate`, `apply`, and `create`.
- ALeA logs all learner interactions (keeps data learner-private)
- each interaction updates the learner model function.

ALeA $\widehat{=}$ Data-Driven & AI-enabled Learning Assistance

- **Ingredient 1:** Domain model $\widehat{=}$ knowledge/theory graph
- **Ingredient 2:** Learner model $\widehat{=}$ adding competency estimations
- **Ingredient 3:** A collection of ready-formulated learning objects

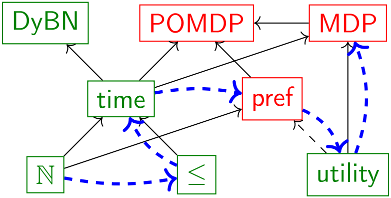


Learning objects are the text fragments learners see and interact with; they are structured by

- didactic relations, e.g. tasks have prerequisites and learning objectives
- rhetoric relations, e.g. introduction, elaboration, and transition

ALeA $\widehat{=}$ Data-Driven & AI-enabled Learning Assistance

- **Ingredient 1:** Domain model $\widehat{=}$ knowledge/theory graph
- **Ingredient 2:** Learner model $\widehat{=}$ adding competency estimations
- **Ingredient 3:** A collection of ready-formulated learning objects
- **Ingredient 4:** Educational dialogue planner $\leadsto$ guided tours

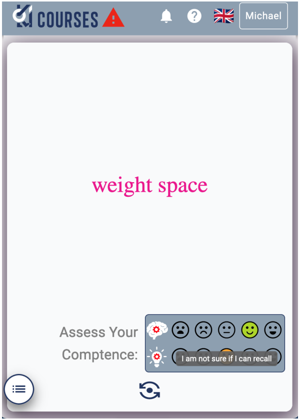


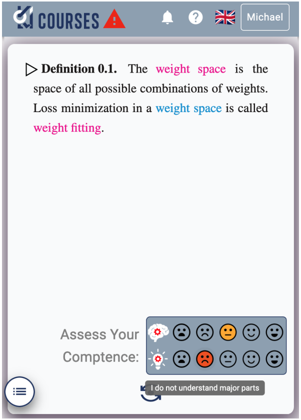
The dialogue planner assembles learning objects into active course materials using

- the domain model and didactic relations to determine the order of LOs
- the learner model to determine what to show
- the rhetoric relations to make the dialogue coherent

New Feature: Drilling with Flashcards

- Flashcards challenge you with a task (term/problem) on the front. . .





▷ **Definition 0.1.** The weight space is the space of all possible combinations of weights. Loss minimization in a weight space is called weight fitting.

. . . and the definition/answer is on the back.

- Self-assessment updates the learner model (before/after)
- **Bonus:** Flashcards can be generated from existing semantic markup (educational equivalent to free beer)

0.2 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

0.2.1 Introduction & Motivation

Knowledge-Representation and -Processing

- **Definition 2.1 (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/is assumed.
- **Definition 2.2.** Knowledge representation formulates knowledge in a formal language so that new knowledge can be induced by inferred via rule systems (inference).
- **Definition 2.3.** We call an information system knowledge based, if a large part of its behaviour is based on inference on represented knowledge.
- **Definition 2.4.** 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 (information visualization)
- 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)

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, . . .)

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, use their intuitions to guide them through only partially understood situations, and can do meta-judgments (moral, practical, . . .)

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, use their intuitions to guide them through only partially understood situations, and can do meta-judgments (moral, practical, . . .)
- **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

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, use their intuitions to guide them through only partially understood situations, and can do meta-judgments (moral, practical, . . .)
- **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.

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, use their intuitions to guide them through only partially understood situations, and can do meta-judgments (moral, practical, . . .)
- **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 2.5.** *More than* 10^5 *math articles published annually in Math.*
- **Observation 2.6.** *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.

The One-Brain-Barrier

- **Observation 2.7.** *More than 10^5 math articles published annually in Math.*
- **Observation 2.8.** *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)

The One-Brain-Barrier

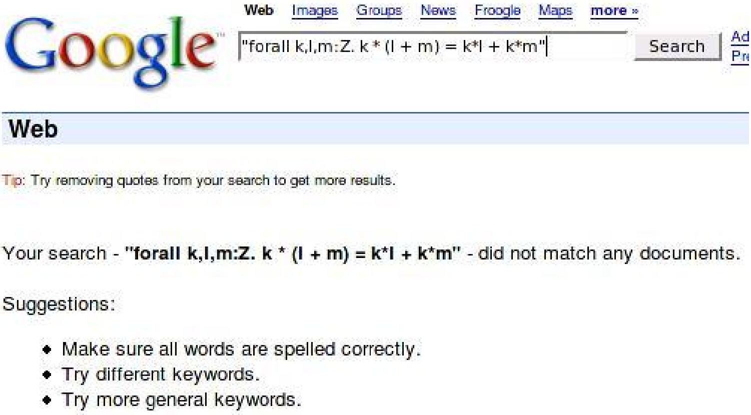
- **Observation 2.9.** *More than 10^5 math articles published annually in Math.*
- **Observation 2.10.** *The libraries of Mizar, Coq, Isabelle,... have $\sim 10^5$ statements+proofs each.* (but are mutually incompatible)
- **Consequence:** Humans lack overview over – let alone working knowledge in – 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)

0.2.2 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 $\mathbb{Z}$ ($\forall k, l, m \in \mathbb{Z}. k \cdot (l + m) = (k \cdot l) + (k \cdot m)$)

Searching for Distributivity



"forall k,l,m:Z. k * (l + m) = k*l + k*m"

Web

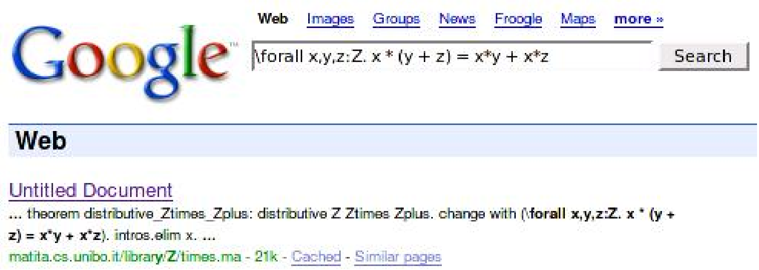
Tip: Try removing quotes from your search to get more results.

Your search - **"forall k,l,m:Z. k * (l + m) = k*l + k*m"** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.

Searching for Distributivity



\forall x,y,z:Z. x * (y + z) = x*y + x*z

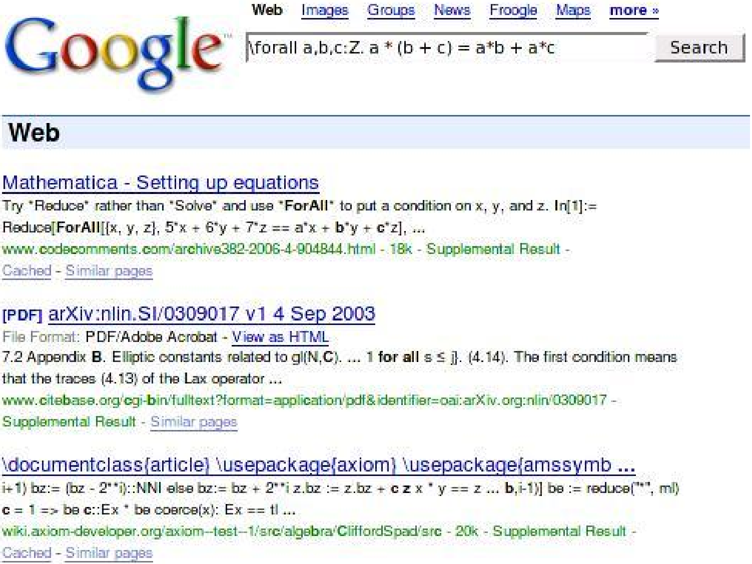
Web

Untitled Document

... theorem distributive_Ztimes_Zplus: distributive Z Ztimes Zplus. change with (**forall x,y,z:Z. x * (y + z) = x*y + x*z**). intros.elim x. ...

matita.cs.unibo.it/library/Z/times.ma - 21k - Cached - Similar pages

Searching for Distributivity



\forall a,b,c:Z. a * (b + c) = a*b + a*c

Web

Mathematica - Setting up equations

Try "Reduce" rather than "Solve" and use "**ForAll**" to put a condition on x, y, and z. In[1]:= Reduce[**ForAll**[{x, y, z}, 5*x + 6*y + 7*z == a*x + **b***y + **c***z], ...

www.codecomments.com/archive382-2006-4-904844.html - 18k - Supplemental Result - Cached - Similar pages

[PDF] arXiv:nlin.SI/0309017 v1 4 Sep 2003

File Format: PDF/Adobe Acrobat - View as HTML

7.2 Appendix **B**. Elliptic constants related to gl(N,**C**). ... 1 **for all** s ≤ j}. (4.14). The first condition means that the traces (4.13) of the Lax operator ...

www.citebase.org/cgi-**b**in/fulltext?format=application/pdf&identifier=oai:arXiv.org:nlin/0309017 - Supplemental Result - Similar pages

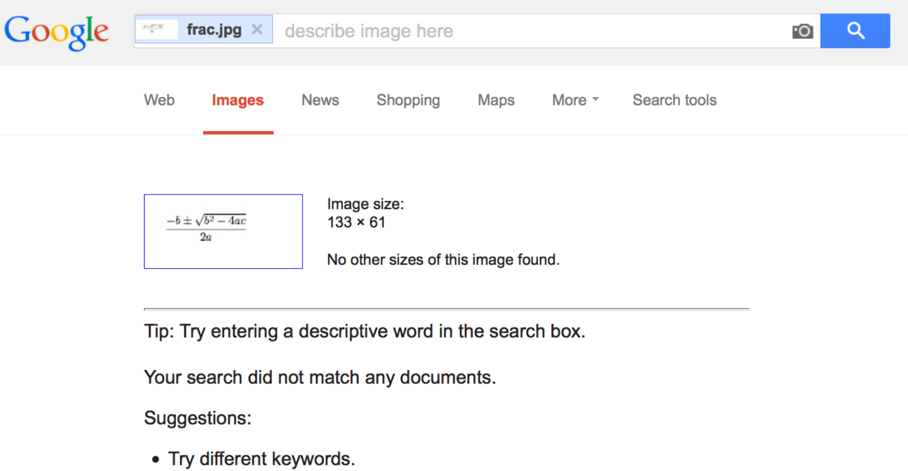
\documentclass{article} \usepackage{axiom} \usepackage{amssymb ...

i+1) bz:= (bz - 2**i)::NNI else bz:= bz + 2**i z.bz := z.bz + **c z** x * y == z ... **b**,i-1)] be := reduce("*", ml) **c** = 1 => be **c**::Ex * be coerce(x): Ex == tl ...

wiki.axiom-developer.org/axiom--test--1/src/algebra/CliffordSpad/src - 20k - Supplemental Result - Cached - Similar pages

Does Image Search help?

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



$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Image size:
133 × 61

No other sizes of this image found.

Tip: Try entering a descriptive word in the search box.

Your search did not match any documents.

Suggestions:

- Try different keywords.

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^n_k, C^k_n, and ${}_k\text{ل}^n$ 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

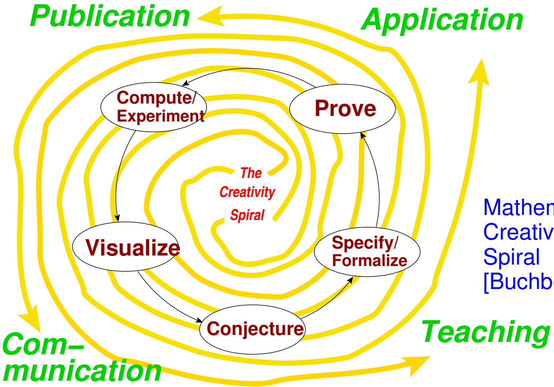
MathWebSearch: Search Math. Formulae on the Web

- **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

0.2.3 The Mathematical Knowledge Space

The way we do math will change dramatically

- **Definition 2.11 (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 2.12 (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 2.13.** We call the skillset that identifies mathematical training mathematical literacy (cf. **??**)

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.

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 2.18.** Understand point sets in 3-space as zeroes of polynomials. Derive insights by studying the algebraic properties of polynomials.
- **Definition 2.19.** We are framing the point sets as algebraic varieties (sets of zeroes of polynomials).

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 2.22.** Understand point sets in 3-space as zeroes of polynomials. Derive insights by studying the algebraic properties of polynomials.
- **Definition 2.23.** We are framing the point sets as algebraic varieties (sets of zeroes of polynomials).
- **Example 2.24 (Lie group).** Equipping a differentiable manifold with a (differentiable) group operation
- **Example 2.25 (Stone's representation theorem).** Interpreting a Boolean algebra as a field of sets.

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 2.26.** Understand point sets in 3-space as zeroes of polynomials. Derive insights by studying the algebraic properties of polynomials.
- **Definition 2.27.** We are framing the point sets as algebraic varieties (sets of zeroes of polynomials).
- **Example 2.28 (Lie group).** Equipping a differentiable manifold with a (differentiable) group operation
- **Example 2.29 (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.

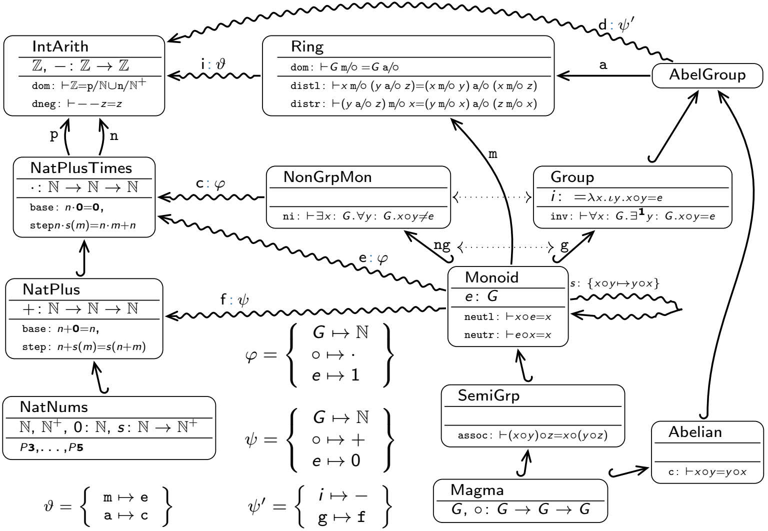
0.2.4 MMT: A Modular Framework for Representing Logics and Domains

Representation language (MMT)

- **Definition 2.30.** MMT $\widehat{=}$ module system for mathematical theories
- Formal syntax and semantics
  - needed for mathematical interface language
  - but how to avoid foundational commitment?
- Foundation-independence
  - identify aspects of underlying language that are necessary for large scale processing
  - formalize exactly those, be parametric in the rest
  - observation: most large scale operations need the same aspects
- Module system
  - preserve mathematical structure wherever possible
  - formal semantics for modularity
- Web-scalable
  - build on XML, OpenMath, OMDoc
  - URI based logical identifiers for all declarations
- Implemented in the MMT API system.

Modular Representation of Math (MMT Example)

- **Example 2.31 (Elementary Algebra and Arithmetics).**



IntArith
$\mathbb{Z}$, $-\colon \mathbb{Z}\to\mathbb{Z}$
dom: $\vdash \mathbb{Z}=\mathtt{p}/\mathbb{N}\cup\mathtt{n}/\mathbb{N}^+$
dneg: $\vdash --z=z$

Ring
dom: $\vdash G\ \mathtt{m}/\circ = G\ \mathtt{a}/\circ$
distl: $\vdash x\ \mathtt{m}/\circ\ (y\ \mathtt{a}/\circ\ z)=(x\ \mathtt{m}/\circ\ y)\ \mathtt{a}/\circ\ (x\ \mathtt{m}/\circ\ z)$
distr: $\vdash (y\ \mathtt{a}/\circ\ z)\ \mathtt{m}/\circ\ x=(y\ \mathtt{m}/\circ\ x)\ \mathtt{a}/\circ\ (z\ \mathtt{m}/\circ\ x)$

AbelGroup

NatPlusTimes
$\cdot\colon \mathbb{N}\to\mathbb{N}\to\mathbb{N}$
base: $n\cdot\mathbf{0}=\mathbf{0}$,
step$n\cdot s(m)=n\cdot m+n$

NonGrpMon
ni: $\vdash\exists x\colon G.\forall y\colon G.x\circ y\neq e$

Group
$i\colon\ =\lambda x.\iota y.x\circ y=e$
inv: $\vdash\forall x\colon G.\exists^{\mathbf{1}}y\colon G.x\circ y=e$

NatPlus
$+\colon \mathbb{N}\to\mathbb{N}\to\mathbb{N}$
base: $n+\mathbf{0}=n$,
step: $n+s(m)=s(n+m)$

Monoid
$e\colon G$
neutl: $\vdash x\circ e=x$
neutr: $\vdash e\circ x=x$

NatNums
$\mathbb{N}$, $\mathbb{N}^+$, $0\colon\mathbb{N}$, $s\colon\mathbb{N}\to\mathbb{N}^+$
$P3,\ldots,P5$

SemiGrp
assoc: $\vdash(x\circ y)\circ z=x\circ(y\circ z)$

Abelian
c: $\vdash x\circ y=y\circ x$

Magma
G, $\circ\colon G\to G\to G$

d : ψ'
a
i : ϑ
p
n
m
c : φ
ng
g
e : φ
f : ψ
s: $\{x\circ y\mapsto y\circ x\}$

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

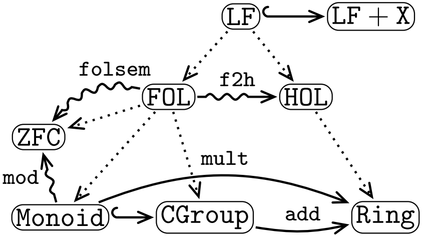
$$\psi=\left\{\begin{array}{l} G\mapsto\mathbb{N}\\ \circ\mapsto +\\ e\mapsto 0\end{array}\right\}$$

$$\vartheta=\left\{\begin{array}{l} \mathtt{m}\mapsto\mathtt{e}\\ \mathtt{a}\mapsto\mathtt{c}\end{array}\right\}$$

$$\psi'=\left\{\begin{array}{l} i\mapsto -\\ \mathtt{g}\mapsto\mathtt{f}\end{array}\right\}$$

Representing Logics and Foundations as Theories

- **Example 2.32.** Logics and foundations represented as MMT theories



- **Definition 2.33.** Meta relation between theories special case of inclusion
- **Uniform Meaning Space:** morphisms between formalizations in different logics become possible via meta-morphisms.
- *Remark 2.34.* Semantics of logics as views into foundations, e.g., `folsem`.
- *Remark 2.35.* Models represented as views into foundations (e.g. ZFC)
- **Example 2.36.** $\mathtt{mod} := \{G \mapsto \mathbb{Z}, \circ \mapsto +, e \mapsto 0\}$ interprets Monoid in $\mathtt{ZFC}$.

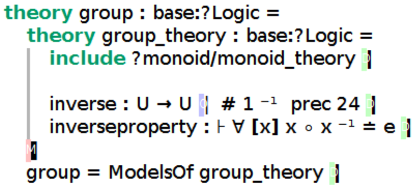
A MitM Theory in MMT Surface Language

- **Example 2.37.** A theory of Groups

Declaration $\widehat{=}$
name : type [= Def] [# notation]

Axioms $\widehat{=}$ Declaration with type $\vdash F$

ModelsOf makes a record type from a theory.



```
theory group : base:?Logic =
  theory group_theory : base:?Logic =
    include ?monoid/monoid_theory ❙

    inverse : U → U ❘ # 1 ⁻¹ prec 24 ❙
    inverseproperty : ⊦ ∀ [x] x ∘ x ⁻¹ ≐ e ❙
  ❚
  group = ModelsOf group_theory ❙
```

- **MitM Foundation:** optimized for natural math formulation
  - higher-order logic based on polymorphic λ-calculus
  - judgements-as-types paradigm: $\vdash F \widehat{=}$ type of proofs of F
  - dependent types with predicate subtyping, e.g. $\{n\}\{'a \in \mathrm{mat}(n,n)|\mathrm{symm}(a)'\}$
  - (dependent) record types for reflecting theories

The MMT Module System

- **Central notion:** theory graph with theory nodes and theory morphisms as edges
- **Definition 2.38.** In MMT, a theory is a sequence of constant declarations optionally with type declarations and definitions
- MMT employs the Curry/Howard isomorphism and treats
  - axioms/conjectures as typed symbol declarations (propositions-as-types)
  - inference rules as function types (proof transformers)
  - theorems as definitions (proof terms for conjectures)
- **Definition 2.39.** MMT had two kinds of theory morphisms
  - structures instantiate theories in a new context (also called: definitional link, import) they import of theory S into theory T induces theory morphism $S \to T$
  - views translate between existing theories (also called: postulated link, theorem link) views transport theorems from source to target (framing).
- Together, structures and views allow a very high degree of re-use
- **Definition 2.40.** We call a statement t induced in a theory T, iff there is
  - a path of theory morphisms from a theory S to T with (joint) assignment σ,
  - such that $t = \sigma(s)$ for some statement s in S.
- **Definition 2.41.** In MMT, all induced statements have a canonical name, the MMT URI.

0.2.5 Application: Serious Games

Framing for Problem Solving (The FrameIT Method)

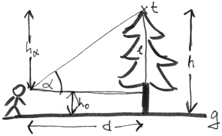
- **Example 2.42 (Problem 0.8.15).**

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

Framing for Problem Solving (The FrameIT Method)

- **Example 2.43 (Problem 0.8.15).**

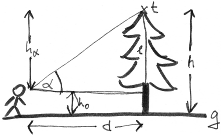
  How can you measure the height of a tree you cannot climb, when you only have a protactor and a tape measure at hand.



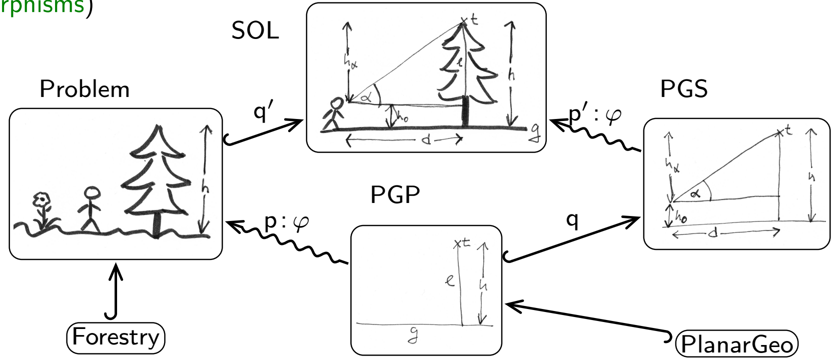
Framing for Problem Solving (The FrameIT Method)

- **Example 2.44 (Problem 0.8.15).**

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



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



SOL

Problem

q′

p′ : φ

PGS

PGP

p : φ

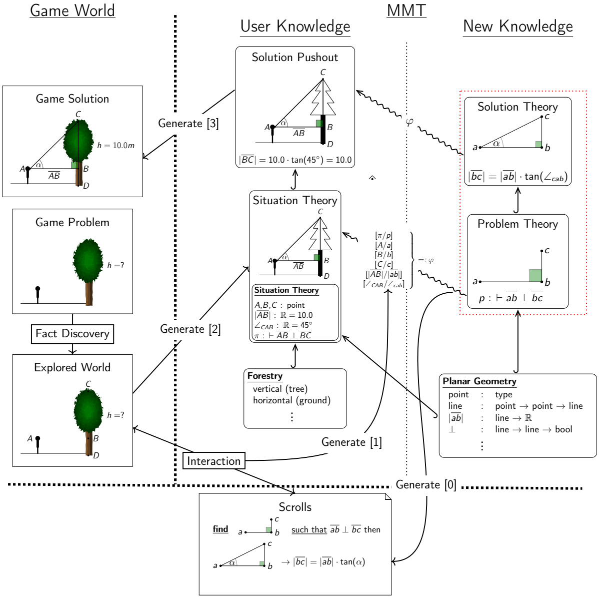
q

Forestry

PlanarGeo

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

Example Learning Object Graph



Game World

MMT

User Knowledge

New Knowledge

Game Solution

Game Problem

Fact Discovery

Explored World

Solution Pushout

$|\overline{BC}| = 10.0 \cdot \tan(45°) = 10.0$

Situation Theory

Situation Theory

A,B,C : point
$|\overline{AB}|$: $\mathbb{R} = 10.0$
$\angle_{CAB}$: $\mathbb{R} = 45°$
$\pi : \vdash \overline{AB} \perp \overline{BC}$

Forestry

vertical (tree)
horizontal (ground)
⋮

Solution Theory

$|\overline{bc}| = |\overline{ab}| \cdot \tan(\angle_{cab})$

Problem Theory

$p : \vdash \overline{ab} \perp \overline{bc}$

Planar Geometry

point	:	type
line	:	point → point → line
$\lvert\overline{ab}\rvert$	:	line → $\mathbb{R}$
$\perp$	:	line → line → bool
⋮		

$$\left.\begin{array}{c}[\pi/p]\\ [A/a]\\ [B/b]\\ [C/c]\\ [|\overline{AB}|/|\overline{ab}|]\\ [\angle_{CAB}/\angle_{cab}]\end{array}\right\} =: \varphi$$

φ

Generate [3]

Generate [2]

Generate [1]

Generate [0]

Interaction

Scrolls

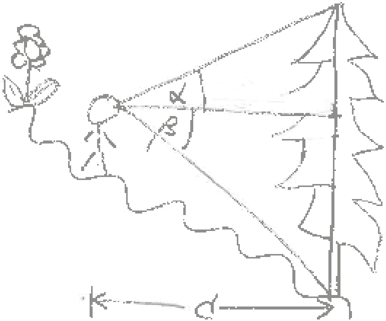
find such that $\overline{ab} \perp \overline{bc}$ then

$\rightarrow |\overline{bc}| = |\overline{ab}| \cdot \tan(\alpha)$

FrameIT Method: Problem

- Problem Representation in the game world (what the student should see) Watch
- Student can interact with the environment via gadgets so solve problems
- “Scrolls” of mathematical knowledge give hints.

Combining Problem/Solution Pairs

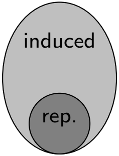


- We can use the same mechanism for combining P/S pairs
- create more complex P/S pairs (e.g. for trees on slopes)

0.2.6 Search in the Mathematical Knowledge Space

The Mathematical Knowledge Space

- **Observation 2.45.** *The value of framing is that it induces new knowledge*
- **Definition 2.46.** 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

♭search: Indexing flattened Theory Graphs

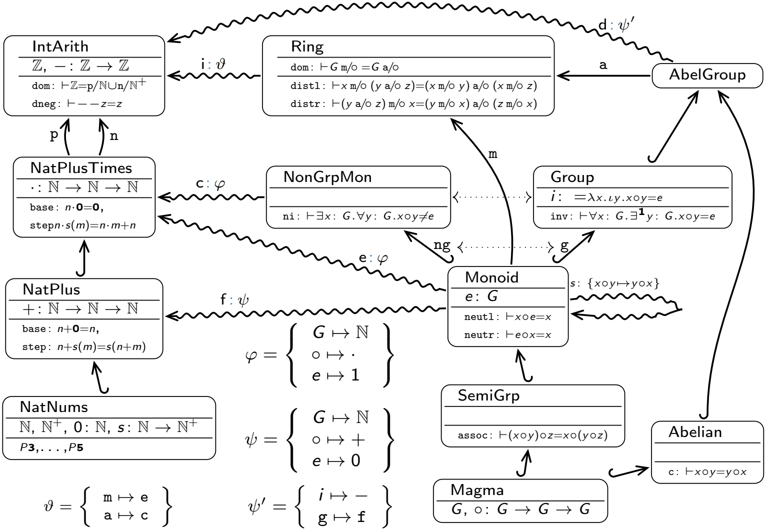
- **Simple Idea:** We have all the necessary components: MMT and `MathWebSearch`
- **Definition 2.47.** The ♭search systen is an integration of `MathWebSearch` and MMT that
  - computes the induced formulae of a modular mathematical library via MMT (aka. flattening)
  - indexes induced formulae by their MMT URIs in `MathWebSearch`
  - uses `MathWebSearch` for unification-based querying (hits are MMT URIs)
  - uses the MMT to present MMT URI (compute the actual formula)
  - generates explanations from the MMT URI of hits.
- Implemented by Mihnea Iancu in ca. 10 days (MMT harvester pre-existed)
  - almost all work was spent on improvements of MMT flattening
  - `MathWebSearch` just worked (web service helpful)

♭search User Interface: Explaining MMT URIs

- **Recall:** ♭search (`MathWebSearch` really) returns a MMT URI as a hit.
- **Question:** How to present that to the user? (for his/her greatest benefit)
- **Fortunately:** MMT system can compute induced statements (the hits)
- **Problem:** Hit statement may look considerably different from the induced statement
- **Solution:** Template-based generation of NL explanations from MMT URIs. MMT knows the necessary information from the components of the MMT URI.

Modular Representation of Math (MMT Example)

- **Example 2.48 (Elementary Algebra and Arithmetics).**



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

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

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

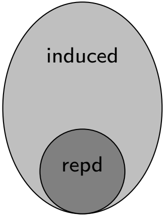
Example: Explaining a MMT URI

- **Example 2.49.** bsearch search result u?IntArith?c/g/assoc for query $(\boxed{x}+\boxed{y})+\boxed{z}=\boxed{R}$.
  - localize the result in the theory u?IntArithf with
    Induced statement $\forall x, y, z : \mathbb{Z}.(x+y)+z = x+(y+z)$ *found in* *http://cds.omdoc.org/cds/elal?IntArith* (*subst*, *justification*).
  - Justification: from MMT info about morphism c (source, target, assignment)
    IntArith is a CGroup if we interpret $\circ$ *as* $+$ *and* G *as* $\mathbb{Z}$.
  - skip over g, since its assignment is trivial and generate
    CGroups are SemiGrps by construction
  - ground the explanation by
    In SemiGrps we have the axiom assoc : $\forall x, y, z : G.(x \circ y) \circ z = x \circ (y \circ z)$

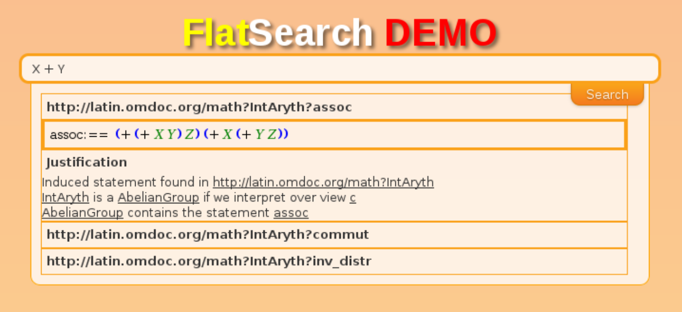
♭search on the LATIN Logic Atlas

- 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`



X + Y

http://latin.omdoc.org/math?IntAryth?assoc

assoc: == (+ (+ X Y) Z) (+ X (+ Y Z))

Justification

Induced statement found in http://latin.omdoc.org/math?IntAryth
IntAryth is a AbelianGroup if we interpret over view c
AbelianGroup contains the statement assoc

http://latin.omdoc.org/math?IntAryth?commut

http://latin.omdoc.org/math?IntAryth?inv_distr

Overview: KWARC Research and Projects

Applications: eMath 3.0, Active Documents, Active Learning, Semantic Spreadsheets/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
- Mathematical Models/Simulation

KM & Interaction:

- Semantic Interpretation (aka. Framing)
- math-literate interaction
- MathHub: math archives & active docs
- Active documents: embedded semantic services
- Model-based Education

Semantization:

- LaTeXML: LaTeX → XML
- sTeX: Semantic LaTeX
- invasive editors
- Context-Aware IDEs
- Mathematical Corpora
- Linguistics of Math
- ML for Math Semantics Extraction

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. (Framing as mathematical practice)

0.3 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
- **Definition 3.1.** Computational Logic is the study of logic from a computational, proof-theoretic perspective. (model theory is mostly comprised under "mathematical logic".)

What is Logic?

- **Definition 3.2.** 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 ($x, y, f, g, p, 1, \pi, \in, \neg, \forall, \exists$)
  - Model: things we understand (e.g. number theory)
  - Interpretation: maps formulae into models ($[\![\text{three plus five}]\!] = 8$)
  - Validity: $\mathcal{M}\models A$, iff $[\![A]\!] = T$ (five greater three is valid)
  - Entailment: $A \models B$, iff $\mathcal{M}\models B$ for all $\mathcal{M}\models A$. (generalize to $\mathcal{H} \models A$)
  - Inference: rules to transform (sets of) formulae ($A, A \Rightarrow B \vdash B$)
  - Syntax: formulae, inference (just a bunch of symbols)
  - Semantics: models, interpr., validity, entailment (math. structures)
- **Important Question:** relation between syntax and semantics?

0.3.1 A History of Ideas in Logic

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-typed 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, semidecidable
  - + set-theoretic semantics ([Tarski 1936])

History of Ideas (continued): Foundations of Mathematics

- Hilbert's Program: find logical system and calculus, ([Hilbert $\sim$ 1930])
  - that formalizes all of mathematics,
  - that admits sound and complete calculi, and
  - whose consistency is provable in the system itself.
- Hilbert's Program is impossible! ([Gödel 1931]) Let $\mathcal{L}$ be a logical system that formalizes arithmetic ($\langle\mathbb{N}, +, *\rangle$),
  - 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

Chapter 1
Foundations of Mathematics

1.1 Propositional Logic and Inference

1.1.1 Propositional Logic (Syntax/Semantics)

Propositional Logic (Syntax)

- **Definition 1.1 (Syntax).** The formulae of propositional logic (write PL^0) are made up from
  - propositional variables: $\mathcal{V}_0 := \{P, Q, R, P^1, P^2, \ldots\}$ (countably infinite)
  - constants/constructors called connectives: $\Sigma_0 := \{T, F, \neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, \ldots\}$

  We define the set $\mathit{wff}_0(\mathcal{V}_0)$ of well-formed propositional formula (wffs) as
  - propositional variables,
  - the logical constants T and F,
  - negations $\neg$A,
  - conjunctions A $\wedge$ B(A and B are called conjuncts),
  - disjunctions A $\vee$ B (A and B are called disjuncts),
  - implications A $\Rightarrow$ B, and
  - equivalences (or biimplication). A $\Leftrightarrow$ B,

  where A, B$\in\mathit{wff}_0(\mathcal{V}_0)$ themselves.
- **Example 1.2.** $P \wedge Q, P \vee Q, (\neg P \vee Q) \Leftrightarrow (P \Rightarrow Q) \in \mathit{wff}_0(\mathcal{V}_0)$
- **Definition 1.3.** Propositional formulae without connectives are called atomic (or an atom) and complex otherwise.

Alternative Notations for Connectives

Here	**Elsewhere**		
$\neg A$	$\sim A$	$\overline{A}$	
$A \wedge B$	$A \mathbin{\&} B$	$A \bullet B$	A, B
$A \vee B$	$A + B$	$A \mid B$	$A ; B$
$A \Rightarrow B$	$A \rightarrow B$	$A \supset B$	
$A \Leftrightarrow B$	$A \leftrightarrow B$	$A \equiv B$	
F	$\bot$	0	
T	$\top$	1	

Semantics of PL^0 (Models)

- **Definition 1.4.** A model $\mathcal{M}:=\langle\mathcal{D}_o,\mathcal{I}\rangle$ for propositional logic consists of
  - the universe $\mathcal{D}_o=\{\mathsf{T},\mathsf{F}\}$
  - the interpretation $\mathcal{I}$ that assigns values to essential connectives.
  - $\mathcal{I}(\neg)\colon\mathcal{D}_o\to\mathcal{D}_o;\mathsf{T}\mapsto\mathsf{F},\mathsf{F}\mapsto\mathsf{T}$
  - $\mathcal{I}(\wedge)\colon\mathcal{D}_o\times\mathcal{D}_o\to\mathcal{D}_o;\langle\alpha,\beta\rangle\mapsto\mathsf{T},\text{ iff }\alpha=\beta=\mathsf{T}$

  We call a constructor a logical constant, iff its value is fixed by the interpretation
- Treat the other connectives as abbreviations, e.g. $\mathsf{A}\vee\mathsf{B}\widehat{=}\neg(\neg\mathsf{A}\wedge\neg\mathsf{B})$ and $\mathsf{A}\Rightarrow\mathsf{B}\widehat{=}\neg\mathsf{A}\vee\mathsf{B}$, and $T\widehat{=}P\vee\neg P$ (only need to treat $\neg,\wedge$ directly)

Semantics of PL^0 (Evaluation)

- **Problem:** The interpretation function only assigns meaning to connectives.
- **Definition 1.5.** A variable assignment $\varphi\colon \mathcal{V}_0 \to \mathcal{D}_o$ assigns values to propositional variables.
- **Definition 1.6.** The value function $\mathcal{I}_\varphi\colon \mathit{wff}_0(\mathcal{V}_0) \to \mathcal{D}_o$ assigns values to PL^0 formulae. It is recursively defined,
  - $\mathcal{I}_\varphi(P) = \varphi(P)$ (base case)
  - $\mathcal{I}_\varphi(\neg \mathrm{A}) = \mathcal{I}(\neg)(\mathcal{I}_\varphi(\mathrm{A}))$.
  - $\mathcal{I}_\varphi(\mathrm{A} \wedge \mathrm{B}) = \mathcal{I}(\wedge)(\mathcal{I}_\varphi(\mathrm{A}), \mathcal{I}_\varphi(\mathrm{B}))$.
- Note that $\mathcal{I}_\varphi(\mathrm{A} \vee \mathrm{B}) = \mathcal{I}_\varphi(\neg(\neg \mathrm{A} \wedge \neg \mathrm{B}))$ is only determined by $\mathcal{I}_\varphi(\mathrm{A})$ and $\mathcal{I}_\varphi(\mathrm{B})$, so we think of the defined connectives as logical constants as well.
- **Definition 1.7.** Two formulae A and B are called equivalent, iff $\mathcal{I}_\varphi(\mathrm{A}) = \mathcal{I}_\varphi(\mathrm{B})$ for all variable assignments φ.

Semantic Properties of Propositional Formulae

- **Definition 1.8.** Let $\mathcal{M}:=\langle\mathcal{U},\mathcal{I}\rangle$ be our model, then we call A
  - true under φ (φ satisfies A) in $\mathcal{M}$, iff $\mathcal{I}_\varphi(\mathrm{A})=\mathsf{T}$ (write $\mathcal{M}\models^\varphi\mathrm{A}$)
  - false under φ (φ falsifies A) in $\mathcal{M}$, iff $\mathcal{I}_\varphi(\mathrm{A})=\mathsf{F}$ (write $\mathcal{M}\not\models^\varphi\mathrm{A}$)
  - satisfiable in $\mathcal{M}$, iff $\mathcal{I}_\varphi(\mathrm{A})=\mathsf{T}$ for some assignment φ
  - valid in $\mathcal{M}$, iff $\mathcal{M}\models^\varphi\mathrm{A}$ for all assignments φ
  - falsifiable in $\mathcal{M}$, iff $\mathcal{I}_\varphi(\mathrm{A})=\mathsf{F}$ for some assignments φ
  - unsatisfiable in $\mathcal{M}$, iff $\mathcal{I}_\varphi(\mathrm{A})=\mathsf{F}$ for all assignments φ
- **Example 1.9.** $x\vee x$ is satisfiable and falsifiable.
- **Example 1.10.** $x\vee\neg x$ is valid and $x\wedge\neg x$ is unsatisfiable.
- **Alternative Notation:** Write $[\![\mathrm{A}]\!]_\varphi$ for $\mathcal{I}_\varphi(\mathrm{A})$, if $\mathcal{M}=\langle\mathcal{U},\mathcal{I}\rangle$. (and $[\![\mathrm{A}]\!]$, if A is ground, and $[\![\mathrm{A}]\!]$, if $\mathcal{M}$ is clear)
- **Definition 1.11 (Entailment).** (aka. logical consequence)
  We say that A entails B ($\mathrm{A}\models\mathrm{B}$), iff $\mathcal{I}_\varphi(\mathrm{B})=\mathsf{T}$ for all φ with $\mathcal{I}_\varphi(\mathrm{A})=\mathsf{T}$ (i.e. all assignments that make A true also make B true)

1.1.2 Calculi for Propositional Logic

Derivation Relations and Inference Rules

- **Definition 1.12.** Let $\mathcal{L}:=\langle\mathcal{L},\mathcal{K},\models\rangle$ be a logical system, then we call a relation $\vdash\subseteq\mathcal{P}(\mathcal{L})\times\mathcal{L}$ a derivation relation for $\mathcal{L}$, if
  - $\mathcal{H}\vdash A$, if $A\in\mathcal{H}$ ($\vdash$ is proof reflexive),
  - $\mathcal{H}\vdash A$ and $\mathcal{H}'\cup\{A\}\vdash B$ imply $\mathcal{H}\cup\mathcal{H}'\vdash B$ ($\vdash$ is proof transitive),
  - $\mathcal{H}\vdash A$ and $\mathcal{H}\subseteq\mathcal{H}'$ imply $\mathcal{H}'\vdash A$ ($\vdash$ is monotonic or admits weakening).
- **Definition 1.13.** We call $\langle\mathcal{L},\mathcal{K},\models,\mathcal{C}\rangle$ a formal system, iff $\mathcal{L}:=\langle\mathcal{L},\mathcal{K},\models\rangle$ is a logical system, and $\mathcal{C}$ a calculus for $\mathcal{L}$.

Derivation Relations and Inference Rules

- **Definition 1.17.** Let $\mathcal{L}:=\langle\mathcal{L},\mathcal{K},\models\rangle$ be a logical system, then we call a relation $\vdash\subseteq\mathcal{P}(\mathcal{L})\times\mathcal{L}$ a derivation relation for $\mathcal{L}$, if
  - $\mathcal{H}\vdash\mathrm{A}$, if $\mathrm{A}\in\mathcal{H}$ ($\vdash$ is proof reflexive),
  - $\mathcal{H}\vdash\mathrm{A}$ and $\mathcal{H}'\cup\{\mathrm{A}\}\vdash\mathrm{B}$ imply $\mathcal{H}\cup\mathcal{H}'\vdash\mathrm{B}$ ($\vdash$ is proof transitive),
  - $\mathcal{H}\vdash\mathrm{A}$ and $\mathcal{H}\subseteq\mathcal{H}'$ imply $\mathcal{H}'\vdash\mathrm{A}$ ($\vdash$ is monotonic or admits weakening).
- **Definition 1.18.** We call $\langle\mathcal{L},\mathcal{K},\models,\mathcal{C}\rangle$ a formal system, iff $\mathcal{L}:=\langle\mathcal{L},\mathcal{K},\models\rangle$ is a logical system, and $\mathcal{C}$ a calculus for $\mathcal{L}$.
- **Definition 1.19.**
  Let $\mathcal{L}$ be the formal language of a logical system, then an inference rule over $\mathcal{L}$ is a decidable $n+1$ ary relation on $\mathcal{L}$. Inference rules are traditionally written as

  $$\frac{\mathrm{A}_1\quad\ldots\quad\mathrm{A}_n}{\mathrm{C}}\mathcal{N}$$

  where $\mathrm{A}_1,\ldots,\mathrm{A}_n$ and C are formula schemata for $\mathcal{L}$ and $\mathcal{N}$ is a name.
  The A_i are called assumptions of $\mathcal{N}$, and C is called its conclusion.
- **Definition 1.20.** An inference rule without assumptions is called an axiom.
- **Definition 1.21.** Let $\mathcal{L}:=\langle\mathcal{L},\mathcal{K},\models\rangle$ be a logical system, then we call a set $\mathcal{C}$ of inference rules over $\mathcal{L}$ a calculus (or inference system) for $\mathcal{L}$.

Derivations

- **Definition 1.22.** Let $\mathcal{L}{:=}\langle\mathcal{L},\mathcal{K},\models\rangle$ be a logical system and $\mathcal{C}$ a calculus for $\mathcal{L}$, then a $\mathcal{C}$-derivation of a formula $\mathrm{C}{\in}\mathcal{L}$ from a set $\mathcal{H}\subseteq\mathcal{L}$ of hypotheses (write $\mathcal{H}{\vdash_{\mathcal{C}}}\mathrm{C}$) is a sequence $\mathrm{A}_1,\ldots,\mathrm{A}_m$ of $\mathcal{L}$-formulae, such that
  - $\mathrm{A}_m = \mathrm{C}$, (derivation culminates in C)
  - for all $1{\leq}i{\leq}m$, either $\mathrm{A}_i{\in}\mathcal{H}$, or (hypothesis)
  - there is an inference rule $\dfrac{\mathrm{A}_{l_1} \quad \ldots \quad \mathrm{A}_{l_k}}{\mathrm{A}_i}$ in $\mathcal{C}$ with $l_j{<}i$ for all $j{\leq}k$. (rule application)

  We can also see a derivation as a derivation tree, where the A_{l_j} are the children of the node A_k.

- **Example 1.23.**

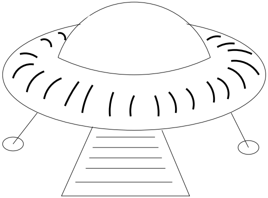
In the propositional Hilbert calculus $\mathcal{H}^0$ we have the derivation $P{\vdash_{\mathcal{H}^0}}Q\Rightarrow P$: the sequence is $P\Rightarrow Q\Rightarrow P, P, Q\Rightarrow P$ and the corresponding tree on the right.

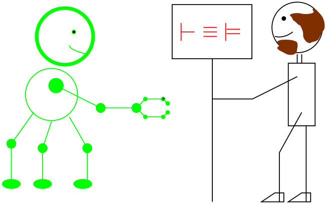
$$\dfrac{\dfrac{}{P\Rightarrow Q\Rightarrow P}\,K \qquad P}{Q\Rightarrow P}\,MP$$

Formal Systems

- Let $\langle\mathcal{L},\mathcal{K},\models\rangle$ be a logical system and $\mathcal{C}$ a calculus, then $\vdash_{\mathcal{C}}$ is a derivation relation and thus $\langle\mathcal{L},\mathcal{K},\models,\vdash_{\mathcal{C}}\rangle$ a derivation system.
- Therefore we will sometimes also call $\langle\mathcal{L},\mathcal{K},\models,\mathcal{C}\rangle$ a formal system, iff $\mathcal{L}{:=}\langle\mathcal{L},\mathcal{K},\models\rangle$ is a logical system, and $\mathcal{C}$ a calculus for $\mathcal{L}$.
- **Definition 1.24.** Let $\mathcal{C}$ be a calculus, then a $\mathcal{C}$-derivation $\emptyset\vdash_{\mathcal{C}}\mathrm{A}$ is called a proof of A and if one exists (write $\vdash_{\mathcal{C}}\mathrm{A}$) then A is called a $\mathcal{C}$-theorem.
  Definition 1.25. The act of finding a proof for a formula A is called proving A.
- **Definition 1.26.**
  An inference rule $\mathcal{I}$ is called admissible in a calculus $\mathcal{C}$, if the extension of $\mathcal{C}$ by $\mathcal{I}$ does not yield new theorems.
- **Definition 1.27.** An inference rule $\dfrac{\mathrm{A}_1 \quad \ldots \quad \mathrm{A}_n}{\mathrm{C}}$ is called derivable (or a derived rule) in a calculus $\mathcal{C}$, if there is a $\mathcal{C}$ derivation $\mathrm{A}_1,\ldots,\mathrm{A}_n\vdash_{\mathcal{C}}\mathrm{C}$.
- **Observation 1.28.** *Derivable inference rules are admissible, but not the other way around.*

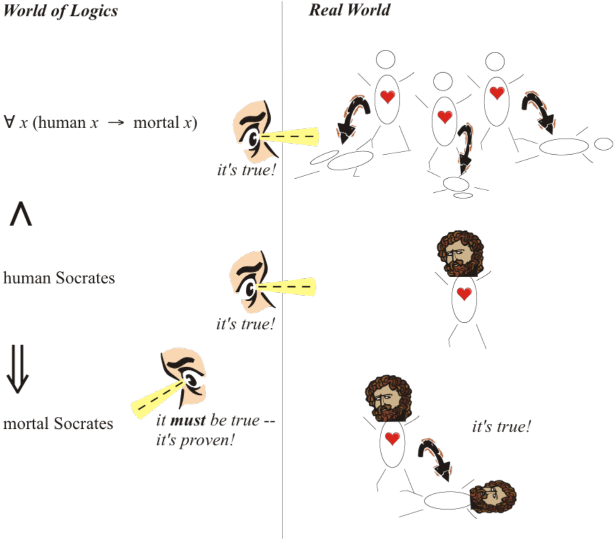
Soundness and Completeness

- **Definition 1.29.** Let $\mathcal{L}{:=}\langle\mathcal{L},\mathcal{K},\models\rangle$ be a logical system, then we call a calculus $\mathcal{C}$ for $\mathcal{L}$, iff
  - sound (or correct), iff $\mathcal{H}\models\mathrm{A}$, whenever $\mathcal{H}\vdash_{\mathcal{C}}\mathrm{A}$, and
  - complete, iff $\mathcal{H}\vdash_{\mathcal{C}}\mathrm{A}$, whenever $\mathcal{H}\models\mathrm{A}$.
- **Goal:** Find calculi C, such that $\vdash_{C}\mathrm{A}$ iff $\models\mathrm{A}$ (provability and validity coincide)
  - To TRUTH through PROOF (CALCULEMUS [Leibniz ∼1680])
  - 



The miracle of logics

- Purely formal derivations are true in the real world!



World of Logics

Real World

$\forall x\,(\text{human } x \rightarrow \text{mortal } x)$

it's true!

$\wedge$

human Socrates

it's true!

$\Downarrow$

mortal Socrates

it ***must*** *be true --*
it's proven!

it's true!

1.1.3 Propositional Natural Deduction Calculus

Calculi: Natural Deduction ($\mathcal{ND}_0$; Gentzen [Gen34])

- **Idea:** $\mathcal{ND}_0$ tries to mimic human argumentation for theorem proving.
- **Definition 1.30.** The propositional natural deduction calculus $\mathcal{ND}_0$ has inference rules for the introduction and elimination of connectives:

Introduction | Elimination | Axiom

$$\frac{A \quad B}{A \wedge B}\wedge I \qquad\qquad \frac{A \wedge B}{A}\wedge E_l \quad \frac{A \wedge B}{B}\wedge E_r \qquad\qquad \frac{}{A \vee \neg A}\,\mathrm{TND}$$

$$\frac{\begin{array}{c}[A]^1\\ \hline\hline \\ B\end{array}}{A \Rightarrow B}\Rightarrow I^1 \qquad\qquad \frac{A \Rightarrow B \quad A}{B}\Rightarrow E$$

$\Rightarrow I$ proves $A \Rightarrow B$ by exhibiting a $\mathcal{ND}_0$ derivation $\mathcal{D}$ (depicted by the double horizontal lines) of B from the local hypothesis A; $\Rightarrow I$ then discharges (get rid of A, which can only be used in $\mathcal{D}$) the hypothesis and concludes $A \Rightarrow B$. This mode of reasoning is called hypothetical reasoning.

- **Definition 1.31.**
  Given a set $\mathcal{H} \subseteq wff_0(\mathcal{V}_0)$ of assumptions and a conclusion C, we write $\mathcal{H} \vdash_{\mathcal{ND}_0} C$, iff there is a $\mathcal{ND}_0$ derivation tree whose leaves are in $\mathcal{H}$.
- **Note:** TND is used only in classical logic (otherwise constructive/intuitionistic)

Natural Deduction: Examples

- **Example 1.32 (Inference with Local Hypotheses).**

$$\dfrac{\dfrac{\dfrac{[\mathrm{A}\wedge\mathrm{B}]^1}{\mathrm{B}}\wedge E_r \qquad \dfrac{[\mathrm{A}\wedge\mathrm{B}]^1}{\mathrm{A}}\wedge E_l}{\mathrm{B}\wedge\mathrm{A}}\wedge I}{\mathrm{A}\wedge\mathrm{B}\Rightarrow\mathrm{B}\wedge\mathrm{A}}\Rightarrow I^1$$

$$\dfrac{\dfrac{\begin{array}{c}[A]^1\\ [B]^2\\ A\end{array}}{B\Rightarrow A}\Rightarrow I^2}{A\Rightarrow B\Rightarrow A}\Rightarrow I^1$$

A Deduction Theorem for $\mathcal{ND}_0$

- **Theorem 1.33.** $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{B}$, *iff* $\mathcal{H} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$.
- *Proof:* We show the two directions separately
  1. If $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{B}$, then $\mathcal{H} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$ by $\Rightarrow I$, and
  2. If $\mathcal{H} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$, then $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$ by weakening and $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{B}$ by $\Rightarrow E$.

More Rules for Natural Deduction

- **Note:** $\mathcal{ND}_0$ does not try to be minimal, but comfortable to work in!x
- **Definition 1.34.** $\mathcal{ND}_0$ has the following additional inference rules for the remaining connectives.

$$\frac{A}{A \vee B}\,\vee I_l \qquad \frac{B}{A \vee B}\,\vee I_r \qquad \frac{A \vee B \quad \begin{array}{c}[A]^1\\ \vdots\\ C\end{array} \quad \begin{array}{c}[B]^1\\ \vdots\\ C\end{array}}{C}\,\vee E^1$$

$$\frac{\begin{array}{c}[A]^1\\ \vdots\\ C\end{array} \quad \begin{array}{c}[A]^1\\ \vdots\\ \neg C\end{array}}{\neg A}\,\neg I^1 \qquad \frac{\neg\neg A}{A}\,\neg E$$

$$\frac{\neg A \quad A}{F}\,FI \qquad \frac{F}{A}\,FE$$

- **Again:** $\neg E$ is used only in classical logic (otherwise constructive/intuitionistic)

Natural Deduction in Sequent Calculus Formulation

- **Idea:** Represent hypotheses explicitly. (lift calculus to judgments)
- **Definition 1.35.** A judgment is a meta statement about the provability of propositions.
- **Definition 1.36.** A sequent is a judgment of the form $\mathcal{H}\vdash A$ about the provability of the formula A from the set $\mathcal{H}$ of hypotheses. We write $\vdash A$ for $\emptyset\vdash A$.
- **Idea:** Reformulate $\mathcal{ND}_0$ inference rules so that they act on sequents.
- **Example 1.37.** We give the sequent style version of 2.35:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}\,Ax}{A \wedge B \vdash B}\,\wedge E_r \qquad \dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}\,Ax}{A \wedge B \vdash A}\,\wedge E_l}{A \wedge B \vdash B \wedge A}\,\wedge I}{\vdash A \wedge B \Rightarrow B \wedge A}\,\Rightarrow I \qquad\qquad \dfrac{\dfrac{\dfrac{}{A, B \vdash A}\,Ax}{A \vdash B \Rightarrow A}\,\Rightarrow I}{\vdash A \Rightarrow B \Rightarrow A}\,\Rightarrow I$$

- **Note:** Even though the antecedent of a sequent is written like a sequence, it is actually a set. In particular, we can permute and duplicate members at will.

Sequent-Style Rules for Natural Deduction

- **Definition 1.38.** The following inference rules make up the propositional sequent style natural deduction calculus $\mathcal{ND}^0_\vdash$:

$$\frac{}{\Gamma, A \vdash A}\,\mathrm{Ax} \qquad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}\,\mathrm{weaken} \qquad \frac{}{\Gamma \vdash A \vee \neg A}\,\mathrm{TND}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,\wedge I \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,\wedge E_l \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,\wedge E_r$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,\vee I_l \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,\vee I_r \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\,\vee E$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\,\Rightarrow I \qquad \frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\,\Rightarrow E$$

$$\frac{\Gamma, A \vdash F}{\Gamma \vdash \neg A}\,\neg I \qquad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\,\neg E$$

$$\frac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash F}\,FI \qquad \frac{\Gamma \vdash F}{\Gamma \vdash A}\,FE$$

Linearized Notation for (Sequent-Style) ND Proofs

- Linearized notation for sequent-style ND proofs

1.	$\mathcal{H}_1$	$\vdash$	A_1	$(\mathcal{J}_1)$
2.	$\mathcal{H}_2$	$\vdash$	A_2	$(\mathcal{J}_2)$
3.	$\mathcal{H}_3$	$\vdash$	A_3	$(\mathcal{J}_3 1, 2)$

corresponds to $$\frac{\mathcal{H}_1 \vdash A_1 \quad \mathcal{H}_2 \vdash A_2}{\mathcal{H}_3 \vdash A_3}\mathcal{R}$$

- **Example 1.39.** We show a linearized version of the $\mathcal{ND}_0$ examples 2.40

#	*hyp*	$\vdash$	*formula*	*NDjust*
1.	1	$\vdash$	$A \wedge B$	Ax
2.	1	$\vdash$	B	$\wedge E_r$ 1
3.	1	$\vdash$	A	$\wedge E_l$ 1
4.	1	$\vdash$	$B \wedge A$	$\wedge I$ 2, 3
5.		$\vdash$	$A \wedge B \Rightarrow B \wedge A$	$\Rightarrow I$ 4

#	*hyp*	$\vdash$	*formula*	*NDjust*
1.	1	$\vdash$	A	Ax
2.	2	$\vdash$	B	Ax
3.	1, 2	$\vdash$	A	weaken 1, 2
4.	1	$\vdash$	$B \Rightarrow A$	$\Rightarrow I$ 3
5.		$\vdash$	$A \Rightarrow B \Rightarrow A$	$\Rightarrow I$ 4

1.2 First-Order Predicate Logic

1.2.1 First-Order Logic

First-Order Predicate Logic (PL^1)

- **Coverage:** We can talk about (*All humans are mortal*)
  - individual things and denote them by variables or constants
  - properties of individuals, (e.g. being human or mortal)
  - relations of individuals, (e.g. *sibling_of* relationship)
  - functions on individuals, (e.g. the *father_of* function)

  We can also state the existence of an individual with a certain property, or the universality of a property.
- But we cannot state assertions like
  - *There is a surjective function from the natural numbers into the reals*.
- First-Order Predicate Logic has many good properties (complete calculi, compactness, unitary, linear unification,...)
- But too weak for formalizing: (at least directly)
  - natural numbers, torsion groups, calculus, ...
  - generalized quantifiers (*most, few,...*)

1.2.1.1 First-Order Logic: Syntax and Semantics

PL^1 Syntax (Signature and Variables)

- **Definition 2.1.**
  First-order logic (PL^1), is a formal system extensively used in mathematics, philosophy, linguistics, and computer science. It combines propositional logic with the ability to quantify over individuals.
- PL^1 talks about two kinds of objects: (so we have two kinds of symbols)
  - truth values by reusing PL^0
  - individuals, e.g. numbers, foxes, Pokémon,. . .
- **Definition 2.2.** A first-order signature consists of (all disjoint; $k\in\mathbb{N}$)
  - connectives: $\Sigma^o = \{T, F, \neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, \ldots\}$ (functions on truth values)
  - function constants: $\Sigma_k^f = \{f, g, h, \ldots\}$ (functions on individuals)
  - predicate constants: $\Sigma_k^p = \{p, q, r, \ldots\}$ (relationships among individuals.)
  - (Skolem constants: $\Sigma_k^{sk} = \{f_k^1, f_k^2, \ldots\}$) (witness constructors; countably ∞)
  - We take Σ_ι to be all of these together: $\Sigma_\iota{:=}\Sigma^f \cup \Sigma^p \cup \Sigma^{sk}$, where $\Sigma^*{:=}\bigcup_{k\in\mathbb{N}}\Sigma_k^*$ and define $\Sigma{:=}\Sigma_\iota \cup \Sigma^o$.
- **Definition 2.3.** We assume a set of individual variables: $\mathcal{V}_\iota{:=}\{X, Y, Z, \ldots\}$. (countably ∞)

PL^1 Syntax (Formulae)

- **Definition 2.4.** Terms: $\mathrm{A} \in wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$ (denote individuals)
  - $\mathcal{V}_\iota \subseteq wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$,
  - if $f \in \Sigma_k^f$ and $\mathrm{A}^i \in wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$ for $i \leq k$, then $f(\mathrm{A}^1, \ldots, \mathrm{A}^k) \in wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$.
- **Definition 2.5.** if Propositions: $\mathrm{A} \in wff_o(\Sigma_\iota, \mathcal{V}_\iota)$: (denote truth values)
  - if $p \in \Sigma_k^p$ and $\mathrm{A}^i \in wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$ for $i \leq k$, then $p(\mathrm{A}^1, \ldots, \mathrm{A}^k) \in wff_o(\Sigma_\iota, \mathcal{V}_\iota)$,
  - if $\mathrm{A}, \mathrm{B} \in wff_o(\Sigma_\iota, \mathcal{V}_\iota)$ and $X \in \mathcal{V}_\iota$, then $T, \mathrm{A} \wedge \mathrm{B}, \neg \mathrm{A}, \forall X.\mathrm{A} \in wff_o(\Sigma_\iota, \mathcal{V}_\iota)$. $\forall$ is a binding operator called the universal quantifier.
- **Definition 2.6.** We define the connectives $F, \vee, \Rightarrow, \Leftrightarrow$ via the abbreviations $\mathrm{A} \vee \mathrm{B} := \neg(\neg \mathrm{A} \wedge \neg \mathrm{B})$, $\mathrm{A} \Rightarrow \mathrm{B} := \neg \mathrm{A} \vee \mathrm{B}$, $\mathrm{A} \Leftrightarrow \mathrm{B} := (\mathrm{A} \Rightarrow \mathrm{B}) \wedge (\mathrm{B} \Rightarrow \mathrm{A})$, and $F := \neg T$. We will use them like the primary connectives $\wedge$ and $\neg$
- **Definition 2.7.** We use $\exists X.\mathrm{A}$ as an abbreviation for $\neg(\forall X.\neg \mathrm{A})$. $\exists$ is a binding operator called the existential quantifier.
- **Definition 2.8.** Call formulae without connectives or quantifiers atomic else complex.

Alternative Notations for Quantifiers

Here	**Elsewhere**	
$\forall x.A$	$\bigwedge x.A$	$(x)A$
$\exists x.A$	$\bigvee x.A$	

Free and Bound Variables

- **Definition 2.9.** We call an occurrence of a variable X bound in a formula A, iff it occurs in a sub-formula $\forall X.\mathrm{B}$ of A. We call a variable occurrence free otherwise.

  For a formula A, we will use $\mathrm{BVar}(\mathrm{A})$ (and $\mathrm{free}(\mathrm{A})$) for the set of bound (free) variables of A, i.e. variables that have a free/bound occurrence in A.
- **Definition 2.10.** We define the set $\mathrm{free}(\mathrm{A})$ of frees variable of a formula A:

$$\begin{array}{l}\mathrm{free}(X){:=}\{X\}\\ \mathrm{free}(f(\mathrm{A}_1,\ldots,\mathrm{A}_n)){:=}\bigcup_{1\leq i\leq n}\mathrm{free}(\mathrm{A}_i)\\ \mathrm{free}(p(\mathrm{A}_1,\ldots,\mathrm{A}_n)){:=}\bigcup_{1\leq i\leq n}\mathrm{free}(\mathrm{A}_i)\\ \mathrm{free}(\neg\mathrm{A}){:=}\mathrm{free}(\mathrm{A})\\ \mathrm{free}(\mathrm{A}\wedge\mathrm{B}){:=}\mathrm{free}(\mathrm{A})\cup\mathrm{free}(\mathrm{B})\\ \mathrm{free}(\forall X.\mathrm{A}){:=}\mathrm{free}(\mathrm{A})\backslash\{X\}\end{array}$$

- **Definition 2.11.** We call a formula A closed or ground, iff $\mathrm{free}(\mathrm{A})=\emptyset$. We call a closed proposition a sentence, and denote the set of all ground terms with $cwff_\iota(\Sigma_\iota)$ and the set of sentences with $cwff_o(\Sigma_\iota)$.

Semantics of PL^1 (Models)

- **Definition 2.12.** We inherit the universe $\mathcal{D}_o = \{\mathsf{T}, \mathsf{F}\}$ of truth values from PL^0 and assume an arbitrary universe $\mathcal{D}_\iota \neq \emptyset$ of individuals(this choice is a parameter to the semantics)
- **Definition 2.13.** An interpretation $\mathcal{I}$ assigns values to constants, e.g.
  - $\mathcal{I}(\neg)\colon \mathcal{D}_o \to \mathcal{D}_o$ with $\mathsf{T} \mapsto \mathsf{F}$, $\mathsf{F} \mapsto \mathsf{T}$, and $\mathcal{I}(\wedge) = \ldots$ (as in PL^0)
  - $\mathcal{I}\colon \Sigma_k^f \to {\mathcal{D}_\iota}^k \to \mathcal{D}_\iota$ (interpret function symbols as arbitrary functions)
  - $\mathcal{I}\colon \Sigma_k^p \to \mathcal{P}({\mathcal{D}_\iota}^k)$ (interpret predicates as arbitrary relations)
- **Definition 2.14.** A variable assignment $\varphi\colon \mathcal{V}_\iota \to \mathcal{D}_\iota$ maps variables into the universe.
- **Definition 2.15.** A model $\mathcal{M} = \langle \mathcal{D}_\iota, \mathcal{I} \rangle$ of PL^1 consists of a universe $\mathcal{D}_\iota$ and an interpretation $\mathcal{I}$.

Semantics of PL^1 (Evaluation)

- **Definition 2.16.**
  Given a model $\langle\mathcal{D},\mathcal{I}\rangle$, the value function $\mathcal{I}_\varphi$ is recursively defined: (two parts: terms & propositions)
  - $\mathcal{I}_\varphi\colon \mathit{wff}_\iota(\Sigma_\iota,\mathcal{V}_\iota)\to\mathcal{D}_\iota$ assigns values to terms.
    - $\mathcal{I}_\varphi(X){:=}\varphi(X)$ and
    - $\mathcal{I}_\varphi(f(\mathrm{A}_1,\ldots,\mathrm{A}_k)){:=}\mathcal{I}(f)(\mathcal{I}_\varphi(\mathrm{A}_1),\ldots,\mathcal{I}_\varphi(\mathrm{A}_k))$
  - $\mathcal{I}_\varphi\colon \mathit{wff}_o(\Sigma_\iota,\mathcal{V}_\iota)\to\mathcal{D}_o$ assigns values to formulae:
    - $\mathcal{I}_\varphi(T)=\mathcal{I}(T)=\mathsf{T}$,
    - $\mathcal{I}_\varphi(\neg\mathrm{A})=\mathcal{I}(\neg)(\mathcal{I}_\varphi(\mathrm{A}))$
    - $\mathcal{I}_\varphi(\mathrm{A}\wedge\mathrm{B})=\mathcal{I}(\wedge)(\mathcal{I}_\varphi(\mathrm{A}),\mathcal{I}_\varphi(\mathrm{B}))$ (just as in PL^0)
    - $\mathcal{I}_\varphi(p(\mathrm{A}_1,\ldots,\mathrm{A}_k)){:=}\mathsf{T}$, iff $\langle\mathcal{I}_\varphi(\mathrm{A}_1),\ldots,\mathcal{I}_\varphi(\mathrm{A}_k)\rangle\in\mathcal{I}(p)$
    - $\mathcal{I}_\varphi(\forall X.\mathrm{A}){:=}\mathsf{T}$, iff $\mathcal{I}_{\varphi,[\mathsf{a}/X]}(\mathrm{A})=\mathsf{T}$ for all $\mathsf{a}\in\mathcal{D}_\iota$.
- **Definition 2.17 (Assignment Extension).** Let φ be a variable assignment into D and $a\in D$, then $\varphi,[a/X]$ is called the extension of φ with $[a/X]$ and is defined as $\{(Y,a)\in\varphi|Y\neq X\}\cup\{(X,a)\}$: $\varphi,[a/X]$ coincides with φ off X, and gives the result a there.

Semantics Computation: Example

- **Example 2.18.** We define an instance of first-order logic:
  - Signature: Let $\Sigma_0^f{:=}\{j, m\}$, $\Sigma_1^f{:=}\{f\}$, and $\Sigma_2^p{:=}\{o\}$
  - Universe: $\mathcal{D}_\iota{:=}\{J, M\}$
  - Interpretation: $\mathcal{I}(j){:=}J$, $\mathcal{I}(m){:=}M$, $\mathcal{I}(f)(J){:=}M$, $\mathcal{I}(f)(M){:=}M$, and $\mathcal{I}(o){:=}\{(M,J)\}$.

  Then $\forall X.o(f(X), X)$ is a sentence and with $\psi{:=}\varphi,[a/X]$ for $a{\in}\mathcal{D}_\iota$ we have

$$
\begin{array}{lll}
\mathcal{I}_\varphi(\forall X.o(f(X), X)) = \mathsf{T} & \text{iff} & \mathcal{I}_\psi(o(f(X), X)) = \mathsf{T} \text{ for all } a{\in}\mathcal{D}_\iota \\
 & \text{iff} & (\mathcal{I}_\psi(f(X)),\mathcal{I}_\psi(X)){\in}\mathcal{I}(o) \text{ for all } a{\in}\{J, M\} \\
 & \text{iff} & (\mathcal{I}(f)(\mathcal{I}_\psi(X)),\psi(X)){\in}\{(M,J)\} \text{ for all } a{\in}\{J, M\} \\
 & \text{iff} & (\mathcal{I}(f)(\psi(X)),a) = (M,J) \text{ for all } a{\in}\{J, M\} \\
 & \text{iff} & \mathcal{I}(f)(a) = M \text{ and } a = J \text{ for all } a{\in}\{J, M\}
\end{array}
$$

  But $a \neq J$ for $a = M$, so $\mathcal{I}_\varphi(\forall X.o(f(X), X)) = \mathsf{F}$ in the model $\langle\mathcal{D}_\iota,\mathcal{I}\rangle$.

1.2.1.2 First-Order Substitutions

Substitutions on Terms

- **Intuition:** If B is a term and X is a variable, then we denote the result of systematically replacing all occurrences of X in a term A by B with $[\mathrm{B}/X](\mathrm{A})$.
- **Problem:** What about $[Z/Y],[Y/X](X)$, is that Y or Z?
- **Folklore:** $[Z/Y],[Y/X](X)=Y$, but $[Z/Y]([Y/X](X))=Z$ of course. (Parallel application)
- **Definition 2.19.**[for=sbstListfromto,sbstListdots,sbst] Let $wfe(\Sigma,\mathcal{V})$ be an expression language, then we call $\sigma\colon\mathcal{V}\to wfe(\Sigma,\mathcal{V})$ a substitution, iff the support $\mathrm{supp}(\sigma){:=}\{X|(X,\mathrm{A}){\in}\sigma,X\neq\mathrm{A}\}$ of σ is finite. We denote the empty substitution with ϵ.
- **Definition 2.20 (Substitution Application).** We define substitution application by
  - $\sigma(c)=c$ for $c{\in}\Sigma$
  - $\sigma(X)=\mathrm{A}$, iff $\mathrm{A}{\in}\mathcal{V}$ and $(X,\mathrm{A}){\in}\sigma$.
  - $\sigma(f(\mathrm{A}_1,\ldots,\mathrm{A}_n))=f(\sigma(\mathrm{A}_1),\ldots,\sigma(\mathrm{A}_n))$,
  - $\sigma(\beta X.\mathrm{A})=\beta X.\sigma_{-X}(\mathrm{A})$.
- **Example 2.21.** $[a/x],[f(b)/y],[a/z]$ instantiates $g(x,y,h(z))$ to $g(a,f(b),h(a))$.
- **Definition 2.22.** Let σ be a substitution then we call $\mathrm{intro}(\sigma){:=}\bigcup_{X\in\mathrm{supp}(\sigma)}\mathrm{free}(\sigma(X))$ the set of variables introduced by σ.

Substitution Extension

- **Definition 2.23 (Substitution Extension).**
  Let σ be a substitution, then we denote the extension of σ with $[\mathrm{A}/X]$ by $\sigma,[\mathrm{A}/X]$ and define it as $\{(Y,\mathrm{B}){\in}\sigma \mid Y \neq X\} \cup \{(X,\mathrm{A})\}$: $\sigma,[\mathrm{A}/X]$ coincides with σ off X, and gives the result A there.
- **Note:** If σ is a substitution, then $\sigma,[\mathrm{A}/X]$ is also a substitution.
- We also need the dual operation: removing a variable from the support:
- **Definition 2.24.** We can discharge a variable X from a substitution σ by setting $\sigma_{-X}{:=}\sigma,[X/X]$.

Substitutions on Propositions

- **Problem:** We want to extend substitutions to propositions, in particular to quantified formulae: What is $\sigma(\forall X.\mathrm{A})$?
- **Idea:** σ should not instantiate bound variables. ($[\mathrm{A}/X](\forall X.\mathrm{B}) = \forall \mathrm{A}.\mathrm{B}'$ ill-formed)
- **Definition 2.25.** $\sigma(\forall X.\mathrm{A}) := (\forall X.\sigma_{-X}(\mathrm{A}))$.
- **Problem:** This can lead to variable capture: $[f(X)/Y](\forall X.p(X,Y))$ would evaluate to $\forall X.p(X,f(X))$, where the second occurrence of X is bound after instantiation, whereas it was free before.
- **Definition 2.26.** Let $\mathrm{B} \in wff_\iota(\Sigma_\iota, \mathcal{V}_\iota)$ and $\mathrm{A} \in wff_o(\Sigma_\iota, \mathcal{V}_\iota)$, then we call B substitutable for X in A, iff A has no occurrence of X in a subterm $\forall Y.\mathrm{C}$ with $Y \in \mathrm{free}(\mathrm{B})$.
- **Solution:** Forbid substitution $[\mathrm{B}/X]\mathrm{A}$, when B is not substitutable for X in A.
- **Better Solution:** Rename away the bound variable X in $\forall X.p(X,Y)$ before applying the substitution. (see alphabetic renaming later.)

Substitution Value Lemma for Terms

- **Lemma 2.27.** *Let* A *and* B *be terms, then* $\mathcal{I}_\varphi([\mathsf{B}/X]\mathsf{A}) = \mathcal{I}_\psi(\mathsf{A})$, *where* $\psi = \varphi, [\mathcal{I}_\varphi(\mathsf{B})/X]$*.*
- *Proof:* by induction on the depth of A:
  1. depth=0 *Then* A *is a variable (say* Y*), or constant, so we have three cases*

     1.1. $\mathsf{A} = Y = X$

     1.1.1. then $\mathcal{I}_\varphi([\mathsf{B}/X](\mathsf{A})) = \mathcal{I}_\varphi([\mathsf{B}/X](X)) = \mathcal{I}_\varphi(\mathsf{B}) = \psi(X) = \mathcal{I}_\psi(X) = \mathcal{I}_\psi(\mathsf{A})$.

     1.2. $\mathsf{A} = Y \neq X$

     1.2.1. then $\mathcal{I}_\varphi([\mathsf{B}/X](\mathsf{A})) = \mathcal{I}_\varphi([\mathsf{B}/X](Y)) = \mathcal{I}_\varphi(Y) = \varphi(Y) = \psi(Y) = \mathcal{I}_\psi(Y) = \mathcal{I}_\psi(\mathsf{A})$.

     1.3. A is a constant

     1.3.1. Analogous to the preceding case ($Y \neq X$).

     1.4. This completes the base case (depth = 0).
  2. depth$>$ 0

     2.1. then $\mathsf{A} = f(\mathsf{A}_1, \ldots, \mathsf{A}_n)$ and we have

$$\begin{aligned} \mathcal{I}_\varphi([\mathsf{B}/X](\mathsf{A})) &= \mathcal{I}(f)(\mathcal{I}_\varphi([\mathsf{B}/X](\mathsf{A}_1)), \ldots, \mathcal{I}_\varphi([\mathsf{B}/X](\mathsf{A}_n))) \\ &= \mathcal{I}(f)(\mathcal{I}_\psi(\mathsf{A}_1), \ldots, \mathcal{I}_\psi(\mathsf{A}_n)) \\ &= \mathcal{I}_\psi(\mathsf{A}). \end{aligned}$$

by inductive hypothesis

Substitution Value Lemma for Propositions

- **Lemma 2.28.** *Let* $\mathrm{B}\in wf\!f_\iota(\Sigma_\iota,\mathcal{V}_\iota)$ *be substitutable for* X *in* $\mathrm{A}\in wf\!f_o(\Sigma_\iota,\mathcal{V}_\iota)$, *then* $\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A}))=\mathcal{I}_\psi(\mathrm{A})$, *where* $\psi=\varphi,[\mathcal{I}_\varphi(\mathrm{B})/X]$.
- *Proof:* by induction on the number n of connectives and quantifiers in A
  1. $n=0$
     1.1. then A is an atomic proposition, and we can argue like in the inductive case of the substitution value lemma for terms.
  2. $n>0$ and $\mathrm{A}=\neg\mathrm{B}$ or $\mathrm{A}=\mathrm{C}\circ\mathrm{D}$
     2.1. Here we argue like in the inductive case of the term lemma as well.
  3. $n>0$ and $\mathrm{A}=\forall X.\mathrm{C}$
     3.1. then $\mathcal{I}_\psi(\mathrm{A})=\mathcal{I}_\psi(\forall X.\mathrm{C})=\mathsf{T}$, iff $\mathcal{I}_{\psi,[a/X]}(\mathrm{C})=\mathcal{I}_{\varphi,[a/X]}(\mathrm{C})=\mathsf{T}$, for all $a\in\mathcal{D}_\iota$, which is the case, iff $\mathcal{I}_\varphi(\forall X.\mathrm{C})=\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A}))=\mathsf{T}$.
  4. $n>0$ and $\mathrm{A}=\forall Y.\mathrm{C}$ where $X\neq Y$
     4.1. then $\mathcal{I}_\psi(\mathrm{A})=\mathcal{I}_\psi(\forall Y.\mathrm{C})=\mathsf{T}$, iff $\mathcal{I}_{\psi,[a/Y]}(\mathrm{C})=\mathcal{I}_{\varphi,[a/Y]}([\mathrm{B}/X](\mathrm{C}))=\mathsf{T}$, by inductive hypothesis.
     4.2. So $\mathcal{I}_\psi(\mathrm{A})=\mathcal{I}_\varphi(\forall Y.[\mathrm{B}/X](\mathrm{C}))=\mathcal{I}_\varphi([\mathrm{B}/X](\forall Y.\mathrm{C}))=\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A}))$

1.2.1.3 Alpha-Renaming for First-Order Logic

Alphabetic Renaming

- **Lemma 2.29.** *Bound variables can be renamed: If Y is substitutable for X in* A*, then* $\mathcal{I}_\varphi(\forall X.\mathrm{A}) = \mathcal{I}_\varphi(\forall Y.[Y/X](\mathrm{A}))$
- *Proof:* by the definitions:
  1. $\mathcal{I}_\varphi(\forall X.\mathrm{A}) = \mathsf{T}$, iff
  2. $\mathcal{I}_{\varphi,[a/X]}(\mathrm{A}) = \mathsf{T}$ for all $a \in \mathcal{D}_\iota$, iff
  3. $\mathcal{I}_{\varphi,[a/Y]}([Y/X](\mathrm{A})) = \mathsf{T}$ for all $a \in \mathcal{D}_\iota$, iff (by substitution value lemma)
  4. $\mathcal{I}_\varphi(\forall Y.[Y/X](\mathrm{A})) = \mathsf{T}$.
- **Definition 2.30.** We call two formulae A and B alphabetical variants (or α-equal; write $\mathrm{A} =_\alpha \mathrm{B}$), iff $\mathrm{A} = \forall X.\mathrm{C}$ and $\mathrm{B} = \forall Y.[Y/X](\mathrm{C})$ for some variables X and Y.

Avoiding Variable Capture by Built-in α-renaming

- **Idea:** Given alphabetic renaming, consider alphabetical variants as identical!
- **So:** Bound variable names in formulae are just a representational device. (we rename bound variables wherever necessary)
- **Formally:** Take $cwff_o(\Sigma_\iota)$ (new) to be the quotient set of $cwff_o(\Sigma_\iota)$ (old) modulo $=_\alpha$. (formulae as syntactic representatives of equivalence classes)
- **Definition 2.31 (Capture-Avoiding Substitution Application).** Let σ be a substitution, A a formula, and A' an alphabetical variant of A, such that $\mathrm{intro}(\sigma) \cap \mathrm{BVar}(\mathrm{A}) = \emptyset$. Then $\mathrm{A}_{=_\alpha} = \mathrm{A}'_{=_\alpha}$ and we can define $\sigma(\mathrm{A}_{=_\alpha}) := (\sigma(\mathrm{A}'))_{=_\alpha}$.
- **Notation:** After we have understood the quotient construction, we will neglect making it explicit and write formulae and substitutions with the understanding that they act on quotients.
- **Alternative:**
  Replace variables with numbers in formulae (de Bruijn indices).

Undecidability of First-Order Logic

- **Theorem 2.32.** *Validity in first-order logic is undecidable.*
- *Proof:* We prove this by contradiction
    1. Let us assume that there is a

1.2.2 First-Order Calculi

1.2.2.1 Propositional Natural Deduction Calculus

Calculi: Natural Deduction ($\mathcal{ND}_0$; Gentzen [Gen34])

- **Idea:** $\mathcal{ND}_0$ tries to mimic human argumentation for theorem proving.
- **Definition 2.33.** The propositional natural deduction calculus $\mathcal{ND}_0$ has inference rules for the introduction and elimination of connectives:

Introduction — Elimination — Axiom

$$\frac{A \quad B}{A \wedge B}\wedge I \qquad \frac{A \wedge B}{A}\wedge E_l \qquad \frac{A \wedge B}{B}\wedge E_r \qquad \frac{}{A \vee \neg A}\mathrm{TND}$$

$$\frac{\begin{array}{c}[A]^1\\ \hline\hline B\end{array}}{A \Rightarrow B}\Rightarrow I^1 \qquad \frac{A \Rightarrow B \quad A}{B}\Rightarrow E$$

$\Rightarrow I$ proves $A \Rightarrow B$ by exhibiting a $\mathcal{ND}_0$ derivation $\mathcal{D}$ (depicted by the double horizontal lines) of B from the local hypothesis A; $\Rightarrow I$ then discharges (get rid of A, which can only be used in $\mathcal{D}$) the hypothesis and concludes $A \Rightarrow B$. This mode of reasoning is called hypothetical reasoning.

- **Definition 2.34.**
  Given a set $\mathcal{H} \subseteq \mathit{wff}_0(\mathcal{V}_0)$ of assumptions and a conclusion C, we write $\mathcal{H} \vdash_{\mathcal{ND}_0} C$, iff there is a $\mathcal{ND}_0$ derivation tree whose leaves are in $\mathcal{H}$.
- **Note:** TND is used only in classical logic (otherwise constructive/intuitionistic)

Natural Deduction: Examples

- **Example 2.35 (Inference with Local Hypotheses).**

$$\cfrac{\cfrac{\cfrac{[A\wedge B]^1}{B}\wedge E_r \qquad \cfrac{[A\wedge B]^1}{A}\wedge E_l}{B\wedge A}\wedge I}{A\wedge B\Rightarrow B\wedge A}\Rightarrow I^1 \qquad\qquad \cfrac{\cfrac{\begin{array}{c}[A]^1\\ [B]^2\\ A\end{array}}{B\Rightarrow A}\Rightarrow I^2}{A\Rightarrow B\Rightarrow A}\Rightarrow I^1$$

A Deduction Theorem for $\mathcal{ND}_0$

- **Theorem 2.36.** $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{B}$, *iff* $\mathcal{H} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$.
- *Proof:* We show the two directions separately
  1. If $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{B}$, then $\mathcal{H} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$ by $\Rightarrow I$, and
  2. If $\mathcal{H} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$, then $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{A} \Rightarrow \mathrm{B}$ by weakening and $\mathcal{H}, \mathrm{A} \vdash_{\mathcal{ND}_0} \mathrm{B}$ by $\Rightarrow E$.

More Rules for Natural Deduction

- **Note:** $\mathcal{ND}_0$ does not try to be minimal, but comfortable to work in!x
- **Definition 2.37.** $\mathcal{ND}_0$ has the following additional inference rules for the remaining connectives.

$$\frac{A}{A \vee B}\,\vee I_l \qquad \frac{B}{A \vee B}\,\vee I_r \qquad \frac{A \vee B \quad \begin{array}{c}[A]^1\\ \vdots\\ C\end{array} \quad \begin{array}{c}[B]^1\\ \vdots\\ C\end{array}}{C}\,\vee E^1$$

$$\frac{\begin{array}{c}[A]^1\\ \vdots\\ C\end{array} \quad \begin{array}{c}[A]^1\\ \vdots\\ \neg C\end{array}}{\neg A}\,\neg I^1 \qquad \frac{\neg\neg A}{A}\,\neg E$$

$$\frac{\neg A \quad A}{F}\,FI \qquad \frac{F}{A}\,FE$$

- **Again:** $\neg E$ is used only in classical logic (otherwise constructive/intuitionistic)

Natural Deduction in Sequent Calculus Formulation

- **Idea:** Represent hypotheses explicitly. (lift calculus to judgments)
- **Definition 2.38.** A judgment is a meta statement about the provability of propositions.
- **Definition 2.39.** A sequent is a judgment of the form $\mathcal{H}\vdash A$ about the provability of the formula A from the set $\mathcal{H}$ of hypotheses. We write $\vdash A$ for $\emptyset\vdash A$.
- **Idea:** Reformulate $\mathcal{ND}_0$ inference rules so that they act on sequents.
- **Example 2.40.**We give the sequent style version of 2.35:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\wedge B\vdash A\wedge B}\,Ax}{A\wedge B\vdash B}\,\wedge E_r \qquad \dfrac{\dfrac{}{A\wedge B\vdash A\wedge B}\,Ax}{A\wedge B\vdash A}\,\wedge E_l}{A\wedge B\vdash B\wedge A}\,\wedge I}{\vdash A\wedge B\Rightarrow B\wedge A}\,\Rightarrow I \qquad\qquad \dfrac{\dfrac{\dfrac{}{A, B\vdash A}\,Ax}{A\vdash B\Rightarrow A}\,\Rightarrow I}{\vdash A\Rightarrow B\Rightarrow A}\,\Rightarrow I$$

- **Note:** Even though the antecedent of a sequent is written like a sequence, it is actually a set. In particular, we can permute and duplicate members at will.

Sequent-Style Rules for Natural Deduction

- **Definition 2.41.** The following inference rules make up the propositional sequent style natural deduction calculus $\mathcal{ND}^0_\vdash$:

$$\frac{}{\Gamma, A \vdash A}\,Ax \qquad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}\,\text{weaken} \qquad \frac{}{\Gamma \vdash A \vee \neg A}\,TND$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,\wedge I \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,\wedge E_l \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,\wedge E_r$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,\vee I_l \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,\vee I_r \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\,\vee E$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\,\Rightarrow I \qquad \frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\,\Rightarrow E$$

$$\frac{\Gamma, A \vdash F}{\Gamma \vdash \neg A}\,\neg I \qquad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\,\neg E$$

$$\frac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash F}\,FI \qquad \frac{\Gamma \vdash F}{\Gamma \vdash A}\,FE$$

Linearized Notation for (Sequent-Style) ND Proofs

- Linearized notation for sequent-style ND proofs

  1. $\mathcal{H}_1 \vdash A_1$ $(\mathcal{J}_1)$
  2. $\mathcal{H}_2 \vdash A_2$ $(\mathcal{J}_2)$
  3. $\mathcal{H}_3 \vdash A_3$ $(\mathcal{J}_3 1, 2)$

  corresponds to $$\frac{\mathcal{H}_1 \vdash A_1 \quad \mathcal{H}_2 \vdash A_2}{\mathcal{H}_3 \vdash A_3}\mathcal{R}$$

- **Example 2.42.** We show a linearized version of the $\mathcal{ND}_0$ examples 2.40

#	*hyp*	$\vdash$	*formula*	*NDjust*
1.	1	$\vdash$	$A \wedge B$	Ax
2.	1	$\vdash$	B	$\wedge E_r$ 1
3.	1	$\vdash$	A	$\wedge E_l$ 1
4.	1	$\vdash$	$B \wedge A$	$\wedge I$ 2, 3
5.		$\vdash$	$A \wedge B \Rightarrow B \wedge A$	$\Rightarrow I$ 4

#	*hyp*	$\vdash$	*formula*	*NDjust*
1.	1	$\vdash$	A	Ax
2.	2	$\vdash$	B	Ax
3.	1, 2	$\vdash$	A	weaken 1, 2
4.	1	$\vdash$	$B \Rightarrow A$	$\Rightarrow I$ 3
5.		$\vdash$	$A \Rightarrow B \Rightarrow A$	$\Rightarrow I$ 4

First-Order Natural Deduction ($\mathcal{ND}^1$; Gentzen [Gen34])

- Rules for connectives just as always
- **Definition 2.43 (New Quantifier Rules).** The first-order natural deduction calculus $\mathcal{ND}^1$ extends $\mathcal{ND}_0$ by the following four rules:

$$\frac{\mathrm{A}}{\forall X.\mathrm{A}}\forall I^* \qquad \frac{\forall X.\mathrm{A}}{[\mathrm{B}/X](\mathrm{A})}\forall E$$

$$\frac{[\mathrm{B}/X](\mathrm{A})}{\exists X.\mathrm{A}}\exists I \qquad \frac{\exists X.\mathrm{A} \qquad \begin{array}{c}[[c/X](\mathrm{A})]^1\\ \vdots\\ \mathrm{C}\end{array} \qquad c\in\Sigma_0^{sk} \text{ new}}{\mathrm{C}}\exists E^1$$

* means that A does not depend on any hypothesis in which X is free.

A Complex $\mathcal{ND}^1$ Example

- **Example 2.44.** We prove $\neg(\forall X.P(X)) \vdash_{\mathcal{ND}^1} \exists X.\neg P(X)$.

$$\dfrac{\dfrac{\dfrac{\neg(\forall X.P(X)) \qquad \dfrac{\dfrac{\dfrac{\dfrac{[\neg(\exists X.\neg P(X))]^1 \qquad \dfrac{[\neg P(X)]^2}{\exists X.\neg P(X)}\,\exists I}{F}\,FI}{\neg\neg P(X)}\,\neg I^2}{P(X)}\,\neg E}{\forall X.P(X)}\,\forall I}{F}\,FI}{\neg\neg(\exists X.\neg P(X))}\,\neg I^1}{\exists X.\neg P(X)}\,\neg E$$

First-Order Natural Deduction in Sequent Formulation

- Rules for connectives from $\mathcal{ND}^0_\vdash$
- **Definition 2.45 (New Quantifier Rules).** The inference rules of the first-order sequent calculus $\mathcal{ND}^1_\vdash$ consist of those from $\mathcal{ND}^0_\vdash$ plus the following quantifier rules:

$$\frac{\Gamma\vdash\mathrm{A}\quad X\notin\mathrm{free}(\Gamma)}{\Gamma\vdash\forall X.\mathrm{A}}\forall I \qquad \frac{\Gamma\vdash\forall X.\mathrm{A}}{\Gamma\vdash[\mathrm{B}/X](\mathrm{A})}\forall E$$

$$\frac{\Gamma\vdash[\mathrm{B}/X](\mathrm{A})}{\Gamma\vdash\exists X.\mathrm{A}}\exists I \qquad \frac{\Gamma\vdash\exists X.\mathrm{A}\quad \Gamma,[c/X](\mathrm{A})\vdash\mathrm{C}\quad c\in\Sigma_0^{sk}\text{ new}}{\Gamma\vdash\mathrm{C}}\exists E$$

Natural Deduction with Equality

- **Definition 2.46 (First-Order Logic with Equality).** We extend PL^1 with a new logical symbol for equality $= \in \Sigma_2^p$ and fix its semantics to $\mathcal{I}(=) := \{(x,x) | x \in \mathcal{D}_\iota\}$. We call the extended logic first-order logic with equality ($\mathrm{PL}^1_=$)
- We now extend natural deduction as well.
- **Definition 2.47.** For the calculus of natural deduction with equality ($\mathcal{ND}^1_=$) we add the following two rules to $\mathcal{ND}^1$ to deal with equality:

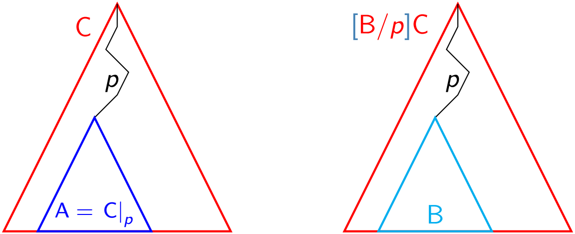
$$\frac{}{\mathrm{A} = \mathrm{A}} {=}I \qquad \frac{\mathrm{A} = \mathrm{B} \quad \mathrm{C}\,[\mathrm{A}]_p}{[\mathrm{B}/p]\mathrm{C}} {=}E$$

  where $\mathrm{C}\,[\mathrm{A}]_p$ if the formula C has a subterm A at position p and $[\mathrm{B}/p]\mathrm{C}$ is the result of replacing that subterm with B.
- In many ways equivalence behaves like equality, we will use the following rules in $\mathcal{ND}^1$
- **Definition 2.48.** $\Leftrightarrow I$ is derivable and $\Leftrightarrow E$ is admissible in $\mathcal{ND}^1$:

$$\frac{}{\mathrm{A} \Leftrightarrow \mathrm{A}} {\Leftrightarrow}I \qquad \frac{\mathrm{A} \Leftrightarrow \mathrm{B} \quad \mathrm{C}\,[\mathrm{A}]_p}{[\mathrm{B}/p]\mathrm{C}} {\Leftrightarrow}E$$

Positions in Formulae

- **Idea:** Formulae are (naturally) trees, so we can use tree positions to talk about subformulae
- **Definition 2.49.** A position p is a tuple of natural numbers that in each node of a expression (tree) specifies into which child to descend. For a expression A we denote the subexpression at p with $\mathrm{A}|_p$.
  We will sometimes write a expression C as $\mathrm{C}\,[\mathrm{A}]_p$ to indicate that C the subexpression A at position p.
- **Definition 2.50.** Let p be a position, then $[\mathrm{A}/p]\mathrm{C}$ is the expression obtained from C by replacing the subexpression at p by A.
- **Example 2.51 (Schematically).**



$\mathcal{ND}^1_=$ Example: $\sqrt{2}$ is Irrational

- We can do real Maths with $\mathcal{ND}^1_=$:
- **Theorem 2.52.** $\sqrt{2}$ *is irrational*

  Proof: We prove the assertion by contradiction

  1. Assume that $\sqrt{2}$ is rational.
  2. Then there are numbers p and q such that $\sqrt{2} = p/q$.
  3. So we know $2q^2 = p^2$.
  4. But $2q^2$ has an odd number of prime factors while p^2 an even number.
  5. This is a contradiction (since they are equal), so we have proven the assertion

$\mathcal{ND}^1_=$ Example: $\sqrt{2}$ is Irrational (the Proof)

#	hyp	formula	NDjust
1		$\forall n, m. \neg(2n+1) = (2m)$	lemma
2		$\forall n, m. \#(n^m) = m\#(n)$	lemma
3		$\forall n, p. \mathrm{prime}(p) \Rightarrow \#(pn) = (\#(n)+1)$	lemma
4		$\forall x. \mathrm{irr}(x) \Leftrightarrow (\neg(\exists p, q. x = p/q))$	definition
5		$\mathrm{irr}(\sqrt{2}) \Leftrightarrow (\neg(\exists p, q. \sqrt{2} = p/q))$	$\forall E(4)$
6	6	$\neg \mathrm{irr}(\sqrt{2})$	Ax
7	6	$\neg\neg(\exists p, q. \sqrt{2} = p/q)$	$\Leftrightarrow E(6,5)$
8	6	$\exists p, q. \sqrt{2} = p/q$	$\neg E(7)$
9	6,9	$\sqrt{2} = p/q$	Ax
10	6,9	$2q^2 = p^2$	arith(9)
11	6,9	$\#(p^2) = 2\#(p)$	$\forall E^2(2)$
12	6,9	$\mathrm{prime}(2) \Rightarrow \#(2q^2) = (\#(q^2)+1)$	$\forall E^2(1)$

$\mathcal{ND}^1_=$ Example: $\sqrt{2}$ is Irrational (the Proof continued)

13		$\mathrm{prime}(2)$	lemma
14	6,9	$\#(2q^2) = \#(q^2) + 1$	$\Rightarrow E(13, 12)$
15	6,9	$\#(q^2) = 2\#(q)$	$\forall E^2(2)$
16	6,9	$\#(2q^2) = 2\#(q) + 1$	$=E(14, 15)$
17		$\#(p^2) = \#(p^2)$	$=I$
18	6,9	$\#(2q^2) = \#(q^2)$	$=E(17, 10)$
19	6.9	$2\#(q) + 1 = \#(p^2)$	$=E(18, 16)$
20	6.9	$2\#(q) + 1 = 2\#(p)$	$=E(19, 11)$
21	6.9	$\neg(2\#(q) + 1) = (2\#(p))$	$\forall E^2(1)$
22	6,9	F	$FI(20, 21)$
23	6	F	$\exists E^6(22)$
24		$\neg\neg\mathrm{irr}(\sqrt{2})$	$\neg I^6(23)$
25		$\mathrm{irr}(\sqrt{2})$	$\neg E^2(23)$

1.3 Higher-Order Logic and λ-Calculus

1.3.1 Higher-Order Predicate Logic

Higher-Order Predicate Logic (PLΩ)

- Quantification over functions and Predicates: $\forall P.\exists F.P(a) \lor \neg P(F(a))$
- **Definition 3.1.** Comprehension: (Existence of Functions)
  $\exists F.\forall X.FX = \mathrm{A}$ e.g. $f(x) = 3x^2 + 5x + 7$
- **Definition 3.2.** Extensionality: (Equality of functions and truth values)
  $\forall F.\forall G.(\forall X.FX = GX) \Rightarrow F = G$
  $\forall P.\forall Q.PQ \Leftrightarrow P = Q$
- **Definition 3.3.** Leibniz Equality: (Indiscernability)
  $\mathrm{A} = \mathrm{B}$ for $\forall P.P\mathrm{A} \Rightarrow P\mathrm{B}$

Problems with PLΩ

- **Problem:** Russell's Antinomy: $\forall Q.\mathcal{M}(Q) \Leftrightarrow (\neg Q(Q))$
  - the set $\mathcal{M}$ of all sets that do not contain themselves
  - **Question** Is $\mathcal{M}\in\mathcal{M}$? **Answer** $\mathcal{M}\in\mathcal{M}$ iff $\mathcal{M}\notin\mathcal{M}$.
- What has happened? the predicate Q has been applied to itself
- **Solution for this course:** Forbid self-applications by types!!
  - ι, prop (type of individuals, truth values), $\alpha \to \beta$ (function type)
  - right associative bracketing: $\alpha \to \beta \to \gamma$ abbreviates $\alpha \to \beta \to \gamma$
  - vector notation: $\overline{\alpha}_n \to \beta$ abbreviates $\alpha_1 \to \ldots \to \alpha_n \to \beta$
- Well-typed formulae (prohibits paradoxes like $\forall Q.\mathcal{M}(Q) \Leftrightarrow (\neg Q(Q))$)
- **Other solution:** Give it a non-standard semantics (Domain-Theory [Scott])

Types

- Types are semantic annotations for terms that prevent antinomies
- **Definition 3.4.** Given a set $\mathcal{BT}$ of base types, construct function types: $\alpha \to \beta$ is the type of functions with domain type α and range type β. We call the closure $\mathcal{T}$ of $\mathcal{BT}$ under function types the set of types over $\mathcal{BT}$.
- **Definition 3.5.**
  We will use ι for the type ofindividuals and prop for the type of truth values.
- **Right Associativity:** The type constructor is used as a right-associative operator, i.e. we use $\alpha \to \beta \to \gamma$ as an abbreviation for $\alpha \to \beta \to \gamma$
- **Vector Notation:**
  We will use a kind of vector notation for function types, abbreviating $\alpha_1 \to \ldots \to \alpha_n \to \beta$ with $\overline{\alpha}_n \to \beta$.

Well-Typed Formulae (PLΩ)

- **Definition 3.6.** Signature $\Sigma_{\mathcal{T}} = \bigcup_{\alpha\in\mathcal{T}}\Sigma_\alpha$ with
- **Definition 3.7.** Connectives: $\neg\in\Sigma_{\text{prop}\to\text{prop}}$ $\{\vee,\wedge,\Rightarrow,\Leftrightarrow,\ldots\}\subseteq\Sigma_{\text{prop}\to\text{prop}\to\text{prop}}$
- **Definition 3.8.** Variables $\mathcal{V}_{\mathcal{T}} = \bigcup_{\alpha\in\mathcal{T}}\mathcal{V}_\alpha$, such that every $\mathcal{V}_\alpha$ countably infinite.
- **Definition 3.9.** Well typed formulae $\mathit{wff}_\alpha(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$ of type α
  - $\mathcal{V}_\alpha\cup\Sigma_\alpha\subseteq\mathit{wff}_\alpha(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$
  - If $\mathrm{C}\in\mathit{wff}_{\alpha\to\beta}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$ and $\mathrm{A}\in\mathit{wff}_\alpha(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$, then $\mathrm{C}\ \mathrm{A}\in\mathit{wff}_\beta(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$
  - If $\mathrm{A}\in\mathit{wff}_{\text{prop}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$, then $\forall X_\alpha.\mathrm{A}\in\mathit{wff}_{\text{prop}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$
- first-order terms have type ι, propositions the type prop.
- there is no type annotation such that $\forall Q.\mathcal{M}(Q)\Leftrightarrow(\neg Q(Q))$ is well-typed. Q needs type α as well as $\alpha\to\text{prop}$.

Standard Semantics for PLΩ

- **Definition 3.10.** The universe of discourse (also carrier) consists of:
  - an arbitrary, non-empty set of individuals $\mathcal{D}_\iota$,
  - a fixed set of truth values $\mathcal{D}_{\mathsf{prop}} = \{\mathsf{T}, \mathsf{F}\}$, and
  - function universes $\mathcal{D}_{(\alpha \to \beta)} = \mathcal{D}_\alpha \to \mathcal{D}_\beta$.
- **Definition 3.11.** Interpretation of constants: typed mapping $\mathcal{I}\colon \Sigma_{\mathcal{T}} \to D$ (i.e. $\mathcal{I}(\Sigma_\alpha) \subseteq \mathcal{D}_\alpha$)
- **Definition 3.12.** We call a structure $\langle \mathcal{D}, \mathcal{I} \rangle$, where $\mathcal{D}$ is a universe and $\mathcal{I}$ an interpretation a standard model of PLΩ.
- **Definition 3.13.** A variable assignment is a typed mapping $\varphi\colon \mathcal{V}_{\mathcal{T}} \to D$.
- **Definition 3.14.** A value function is a typed mapping $\mathcal{I}_\varphi\colon \mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}}) \to D$ with
  - $\mathcal{I}_\varphi|_{\mathcal{V}_{\mathcal{T}}} = \varphi \qquad \mathcal{I}_\varphi|_{\Sigma_{\mathcal{T}}} = \mathcal{I}$
  - $\mathcal{I}_\varphi(\mathsf{A}\,\mathsf{B}) = \mathcal{I}_\varphi(\mathsf{A})(\mathcal{I}_\varphi(\mathsf{B}))$
  - $\mathcal{I}_\varphi(\forall X_\alpha.\mathsf{A}) = \mathsf{T}$, iff $\mathcal{I}_{\varphi,[\mathsf{a}/X]}(\mathsf{A}) = \mathsf{T}$ for all $\mathsf{a} \in \mathcal{D}_\alpha$.
- **Definition 3.15.** $\mathsf{A}_{\mathsf{prop}}$ valid under φ, iff $\mathcal{I}_\varphi(\mathsf{A}) = \mathsf{T}$.

Equality

- **Definition 3.16 (Leibniz equality).** $\mathrm{Q}^{\alpha}\mathrm{A}_{\alpha}\mathrm{B}_{\alpha} = \forall P_{\alpha\to\mathrm{prop}}.PA \Leftrightarrow PB$ (indiscernability)
- **Note:** $\forall P_{\alpha\to\mathrm{prop}}.PA \Rightarrow PB$ (get the other direction by instantiating P with Q, where $QX \Leftrightarrow (\neg PX)$)
- **Theorem 3.17.** *If $\mathcal{M} = \langle\mathcal{D},\mathcal{I}\rangle$ is a standard model, then $\mathcal{I}_{\varphi}(\mathrm{Q}^{\alpha})$ is the identity relation on $\mathcal{D}_{\alpha}$.*
- **Notation:** We write $\mathrm{A} = \mathrm{B}$ for QAB(A and B are equal, iff there is no property P that can tell them apart.)
- *Proof:*
  1. $\mathcal{I}_{\varphi}(\mathrm{QAB}) = \mathcal{I}_{\varphi}(\forall P.PA \Rightarrow PB) = \mathsf{T}$, iff $\mathcal{I}_{\varphi,[r/P]}(PA \Rightarrow PB) = \mathsf{T}$ for all $r{\in}\mathcal{D}_{(\alpha\to\mathrm{prop})}$.
  2. For $\mathrm{A} = \mathrm{B}$ we have $\mathcal{I}_{\varphi,[r/P]}(PA) = r(\mathcal{I}_{\varphi}(\mathrm{A})) = \mathsf{F}$ or $\mathcal{I}_{\varphi,[r/P]}(PB) = r(\mathcal{I}_{\varphi}(\mathrm{B})) = \mathsf{T}$.
  3. Thus $\mathcal{I}_{\varphi}(\mathrm{QAB}) = \mathsf{T}$.
  4. Let $\mathcal{I}_{\varphi}(\mathrm{A}) \neq \mathcal{I}_{\varphi}(\mathrm{B})$ and $r{=}\{\mathcal{I}_{\varphi}(\mathrm{A})\}{\in}\mathcal{D}_{(\alpha\to\mathrm{prop})}$ (exists in a standard model)
  5. so $r(\mathcal{I}_{\varphi}(\mathrm{A})) = \mathsf{T}$ and $r(\mathcal{I}_{\varphi}(\mathrm{B})) = \mathsf{F}$
  6. $\mathcal{I}_{\varphi}(\mathrm{QAB}) = \mathsf{F}$, as $\mathcal{I}_{\varphi,[r/P]}(PA \Rightarrow PB) = \mathsf{F}$, since $\mathcal{I}_{\varphi,[r/P]}(PA) = r(\mathcal{I}_{\varphi}(\mathrm{A})) = \mathsf{T}$ and $\mathcal{I}_{\varphi,[r/P]}(PB) = r(\mathcal{I}_{\varphi}(\mathrm{B})) = \mathsf{F}$.

Example: Peano Axioms for the Natural Numbers

- $\Sigma_{\mathcal{T}} = \{[\mathbb{N}{:}\iota \to \mathsf{prop}], [0{:}\iota], [s{:}\iota \to \iota]\}$
- $\mathbb{N}0$ (0 is a natural number)
- $\forall X_\iota.\mathbb{N}X \Rightarrow \mathbb{N}(sX)$ (the successor of a natural number is natural)
- $\neg(\exists X_\iota.\mathbb{N}X \wedge sX = 0)$ (0 has no predecessor)
- $\forall X_\iota.\forall Y_\iota.(sX = sY) \Rightarrow X = Y$ (the successor function is injective)
- $\forall P_{\iota\to\mathsf{prop}}.P0 \Rightarrow (\forall X_\iota.\mathbb{N}X \Rightarrow PX \Rightarrow P(sX)) \Rightarrow (\forall.\mathbb{N}Y \Rightarrow P(Y))$
  induction axiom: all properties P, that hold of 0, and with every n for its successor $s(n)$, hold on all $\mathbb{N}$

Expressive Formalism for Mathematics

- **Example 3.18 (Cantor's Theorem).** The cardinality of a set is smaller than that of its power set.
  - smaller-card(M,N):=$\neg(\exists F.\mathrm{surjective}(F,M,N))$
  - surjective(F,M,N):=$(\forall X{\in}M.\exists Y{\in}N.FY=X)$
- **Example 3.19 (Simplified Formalization).** $\neg(\exists F_{\iota\to\iota\to\iota}.\forall G_{\iota\to\iota}.\exists J_{\iota}.FJ=G)$
- Standard-Benchmark for higher-order theorem provers
- can be proven by Tps and Leo (see below)

Hilbert-Calculus

- **Definition 3.20 ($\mathcal{H}_\Omega$ Axioms).**
  - $\forall P_{\mathrm{prop}}, Q_{\mathrm{prop}}.P \Rightarrow Q \Rightarrow P$
  - $\forall P_{\mathrm{prop}}, Q_{\mathrm{prop}}, R_{\mathrm{prop}}.(P \Rightarrow Q \Rightarrow R) \Rightarrow (P \Rightarrow Q) \Rightarrow P \Rightarrow R$
  - $\forall P_{\mathrm{prop}}, Q_{\mathrm{prop}}.(\neg P \Rightarrow \neg Q) \Rightarrow P \Rightarrow Q$
- **Definition 3.21 ($\mathcal{H}_\Omega$ inference rules).**

$$\frac{\mathrm{A}_{\mathrm{prop}} \Rightarrow \mathrm{B}_{\mathrm{prop}} \quad \mathrm{A}}{\mathrm{B}} \qquad \frac{\forall X_\alpha.\mathrm{A}}{[\mathrm{B}/X_\alpha](\mathrm{A})} \qquad \frac{\mathrm{A}}{\forall X_\alpha.\mathrm{A}} \qquad \frac{X \notin \mathrm{free}(\mathrm{A}) \quad \forall X_\alpha.\mathrm{A} \wedge \mathrm{B}}{\mathrm{A} \wedge (\forall X_\alpha.\mathrm{B})}$$

- **Theorem 3.22.** *Sound, wrt. standard semantics*
- Also Complete?

Hilbert-Calculus $\mathcal{H}_\Omega$ (continued)

- **Example 3.23.** Valid sentences that are not $\mathcal{H}_\Omega$-theorems:
  - Cantor's Theorem:
    $\neg(\exists F_{\iota\to\iota\to\iota}.\forall G_{\iota\to\iota}.(\forall K_\iota.\mathbb{N}\ K \Rightarrow \mathbb{N}\ G\ K) \Rightarrow (\exists.\mathbb{N}\ J \wedge FJ = G))$
    (There is no surjective mapping from $\mathbb{N}$ into the set $\mathbb{N}\to,\mathbb{N}$ of natural number sequences)
  - proof attempt fails at the subgoal $\exists G_{\iota\to\iota}.\forall X_\iota.GX = s(fXX)$
- **Definition 3.24 (New Axiom Schema).** Comprehension axiom
  $\exists F_{\alpha\to\beta}.\forall X_\alpha.F\ X = \mathrm{A}_\beta$ (for every variable X_α and every term $\mathrm{A}{\in}\mathit{wff}_\beta(\Sigma_\mathcal{T},\mathcal{V}_\mathcal{T})$)
- **Definition 3.25 (new axiom schemata).** Extensionality axiom

$$\begin{array}{ll} \mathrm{Ext}^{\alpha\beta} & \forall F_{\alpha\to\beta}.\forall G_{\alpha\to\beta}.(\forall X_\alpha.FX = GX) \Rightarrow F = G \\ \mathrm{Ext}^{o} & \forall F_{\mathrm{prop}}.\forall G_{\mathrm{prop}}.FG \Leftrightarrow F = G \end{array}$$

- correct!   complete? cannot be!! [Göd31]

Way Out: Henkin-Semantics

- **Observation:** Gödel's incompleteness theorem only holds for standard semantics.
- **Idea:** Find generalization that admits complete calculi
- **Concretely:** Generalize so that the carrier only contains those functions that are requested by the comprehension axioms.
- **Theorem 3.26 (Henkin's theorem).** $\mathcal{H}_\Omega$ *is complete wrt. this semantics.*
- *Proof sketch:* more models $\leadsto$ less valid sentences (these are $\mathcal{H}_\Omega$-theorems)
- Henkin-models induce sensible measure of completeness for higher-order logic.

1.3.2 A better Form of Comprehension and Extensionality

From Comprehension to β-Conversion

- $\exists F_{\alpha\to\beta}.\forall X_\alpha.FX = A_\beta$ for arbitrary variable X_α and term $A\in wff_\beta(\Sigma_\mathcal{T},\mathcal{V}_\mathcal{T})$ (for each term A and each variable X there is a function $f\in\mathcal{D}_{(\alpha\to\beta)}$, with $f(\varphi(X)) = \mathcal{I}_\varphi(A))$
  - schematic in α, β, X_α and A_β, very inconvenient for deduction
- Transformation in $\mathcal{H}_\Omega$
  - $\exists F_{\alpha\to\beta}.\forall X_\alpha.FX = A_\beta$
  - $\forall X_\alpha.(\lambda X_\alpha.A)X = A_\beta$ $(\exists E)$
    Call the function F whose existence is guaranteed "$(\lambda X_\alpha.A)$"
  - $(\lambda X_\alpha.A)B = [B/X]A_\beta$ $(\forall E)$, in particular for $B\in wff_\alpha(\Sigma_\mathcal{T},\mathcal{V}_\mathcal{T})$.
- **Definition 3.27.** Axiom of β equality: $(\lambda X_\alpha.A)\,B = [B/X](A_\beta)$
- **Idea:** Introduce a new class of formulae (λ-calculus [Chu40])

From Extensionality to η-Conversion

- **Definition 3.28.** Extensionality Axiom: $\forall F_{\alpha\to\beta}.\forall G_{\alpha\to\beta}.(\forall X_\alpha.FX = GX) \Rightarrow F = G$
- **Idea:** Maybe we can get by with a simplified equality schema here as well.
- **Definition 3.29.** We say that A and $\lambda X_\alpha.\mathrm{A}\ X$ are η-equal, (write $\mathrm{A}_{\alpha\to\beta} =_\eta (\lambda X_\alpha.\mathrm{A}\ X)$), iff $X \notin \mathrm{free}(\mathrm{A})$.
- **Theorem 3.30.** *η-equality and Extensionality are equivalent*
- *Proof:* We show that η-equality is special case of extensionality; the converse direction is trivial
  1. Let $\forall X_\alpha.\mathrm{A}X = \mathrm{B}X$, thus $\mathrm{A}X = \mathrm{B}X$ with $\forall E$
  2. $\lambda X_\alpha.\mathrm{A}X = \lambda X_\alpha.\mathrm{B}X$, therefore $\mathrm{A} = \mathrm{B}$ with η
  3. Hence $\forall F_{\alpha\to\beta}.\forall G_{\alpha\to\beta}.(\forall X_\alpha.FX = GX) \Rightarrow F = G$ by twice $\forall I$.
- Axiom of truth values: $\forall F_{\mathrm{prop}}.\forall G_{\mathrm{prop}}.FG \Leftrightarrow F = G$ unsolved.

1.3.3 Simply Typed λ-Calculus

Simply typed λ-Calculus (Syntax)

- **Definition 3.31.** Signature $\Sigma_{\mathcal{T}} = \bigcup_{\alpha \in \mathcal{T}} \Sigma_\alpha$ (includes countably infinite signatures Σ_α^{Sk} of Skolem contants).
- $\mathcal{V}_{\mathcal{T}} = \bigcup_{\alpha \in \mathcal{T}} \mathcal{V}_\alpha$, such that $\mathcal{V}_\alpha$ are countably infinite.
- **Definition 3.32.** We call the set $\mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ defined by the rules
  - $\mathcal{V}_\alpha \cup \Sigma_\alpha \subseteq \mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$
  - If $\mathrm{C} \in \mathit{wff}_{\alpha \to \beta}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ and $\mathrm{A} \in \mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$, then $\mathrm{C}\ \mathrm{A} \in \mathit{wff}_\beta(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$
  - If $\mathrm{A} \in \mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$, then $\lambda X_\beta.\mathrm{A} \in \mathit{wff}_{\beta \to \alpha}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$

  the set of well typed formulae of type α over the signature $\Sigma_{\mathcal{T}}$ and use $\mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}}) := \bigcup_{\alpha \in \mathcal{T}} \mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ for the set of all well-typed formulae.
- **Definition 3.33.** We will call all occurrences of the variable X in A bound in $\lambda X.\mathrm{A}$. Variables that are not bound in B are called free in B.
- Substitutions are well typed, i.e. $\sigma(X_\alpha) \in \mathit{wff}_\alpha(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ and capture-avoiding.
- **Definition 3.34 (Simply Typed λ-Calculus).** The simply typed λ calculus $\Lambda^{\to}$ over a signature $\Sigma_{\mathcal{T}}$ has the formulae $\mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ (they are called λ-terms) and the following equalities:
  - α conversion: $(\lambda X.\mathrm{A}) =_\alpha (\lambda Y.[Y/X](\mathrm{A}))$.
  - β conversion: $(\lambda X.\mathrm{A})\ \mathrm{B} =_\beta [\mathrm{B}/X](\mathrm{A})$.
  - η conversion: $(\lambda X.\mathrm{A}\ X) =_\eta \mathrm{A}$ if $X \notin \mathrm{free}(\mathrm{A})$.

Simply typed λ-Calculus (Notations)

- **Application is left-associative:** We abbreviate $\mathrm{F}\ \mathrm{A}^1\ \mathrm{A}^2\ \ldots\ \mathrm{A}^n$ with $\mathrm{F}(\mathrm{A}^1,\ldots,\mathrm{A}^n)$ eliding the brackets and further with $\mathrm{F}\ \overline{\mathrm{A}^n}$ in a kind of vector notation.
- **Andrews' dot Notation:** $\mathrm{A}\ .$ stands for a left bracket whose partner is as far right as is consistent with existing brackets; i.e. $\mathrm{A}\ .\mathrm{B}\ \mathrm{C}$ abbreviates $\mathrm{A}\ (\mathrm{B}\ \mathrm{C})$.
- **Abstraction is right-associative:** We abbreviate $\lambda X^1.\lambda X^2.\cdots\lambda X^n.\mathrm{A}\cdots$ with $\lambda X^1\ldots X^n.\mathrm{A}$ eliding brackets, and further to $\lambda\overline{X^n}.\mathrm{A}$ in a kind of vector notation.
- **Outer brackets:** Finally, we allow ourselves to elide outer brackets where they can be inferred.

$=_{\alpha\beta\eta}$-Equality (Overview)

- reduction with $\left\{\begin{array}{ll} =_\beta : & (\lambda X.A)\,B \to_\beta [B/X](A) \\ =_\eta : & (\lambda X.A\,X) \to_\eta A \end{array}\right.$ under $=_\alpha$: $\begin{array}{c} \lambda X.A \\ =_\alpha \\ \lambda Y.[Y/X](A) \end{array}$
- **Theorem 3.35.** *β-reduction is well-typed, terminating and confluent in the presence of α-conversion.*
- **Definition 3.36 (Normal Form).** We call a λ-term A a normal form (in a reduction system $\mathcal{E}$), iff no rule (from $\mathcal{E}$) can be applied to A.
- **Corollary 3.37.** *$=_{\beta\eta}$-reduction yields unique normal forms (up to $=_\alpha$-equivalence).*

Syntactic Parts of λ-Terms

- **Definition 3.38 (Parts of λ-Terms).**
  We can always write a λ-term in the form $\mathrm{T} = \lambda X^1 \ldots X^k.\mathrm{H}\mathrm{A}^1 \ldots \mathrm{A}^n$, where H is not an application. We call
  - H the syntactic head of T
  - $\mathrm{H}(\mathrm{A}^1, \ldots, \mathrm{A}^n)$ the matrix of T, and
  - $\lambda X^1 \ldots X^k.$ (or the sequence $X^1, \ldots, X^k$) the binder of T
- **Definition 3.39.**
  Head reduction always has a unique β redex

$$(\lambda\overline{X^n}.\lambda Y.\mathrm{A}(\mathrm{B}^2, \ldots, \mathrm{B}^n)) \to_\beta^h (\lambda\overline{X^n}.[\mathrm{B}^1/Y](\mathrm{A})(\mathrm{B}^2, \ldots, \mathrm{B}^n))$$

- **Theorem 3.40.** *The syntactic heads of β-normal forms are constant or variables.*
- **Definition 3.41.** Let A be a λ-term, then the syntactic head of the β-normal form of A is called the head symbol of A and written as $\mathrm{head}(\mathrm{A})$. We call a λ-term a j-projection, iff its head is the j^{th} bound variable.
- **Definition 3.42.** We call a λ-term a η long form, iff its matrix has base type.
- **Definition 3.43.** η Expansion makes η long forms

$$\eta\left[(\lambda X^1 \ldots X^n.\mathrm{A})\right] := (\lambda X^1 \ldots X^n.\lambda Y^1 \ldots Y^m.\mathrm{A}(Y^1, \ldots, Y^m))$$

- **Definition 3.44.** Long $\beta\eta$ normal form, iff it is β normal and η-long.

A Test Generator for Higher-Order Unification

- **Definition 3.45 (Church Numerals).** We define closed λ-terms of type $\nu{:=}\alpha\to\alpha\to\alpha\to\alpha$
  - Numbers: Church numerals: (*n* fold iteration of arg1 starting from arg2)

$$n{:=}(\lambda S_{\alpha\to\alpha}.\lambda O_\alpha.\underbrace{S(S\ldots S}_{n}(O)\ldots))$$

  - Addition (*N*-fold iteration of *S* from *N*)

$$+{:=}(\lambda N_\nu M_\nu.\lambda S_{\alpha\to\alpha}.\lambda O_\alpha.NS(MSO))$$

  - Multiplication: (*N*-fold iteration of *MS* (=+*m*) from *O*)

$$\cdot{:=}(\lambda N_\nu M_\nu.\lambda S_{\alpha\to\alpha}.\lambda O_\alpha.N(MS)O)$$

- **Observation 3.46.** *Subtraction and (integer) division on Church numberals can be automted via higher-order unification.*
- **Example 3.47.**
  $5-2$ by solving the unification problem $(2+x_\nu){=^?}5$
- Equation solving for Church numerals yields a very nice generator for test cases for higher-order unification, as we know which solutions to expect.

1.3.4 Simply Typed λ-Calculus via Inference Systems

Simply Typed λ-Calculus as an Inference System: Terms

- **Idea:** Develop the λ-calculus in two steps
  - A context-free grammar for “raw λ-terms” (for the structure)
  - Identify the well-typed λ-terms in that (cook them until well-typed)
- **Definition 3.48.**
  A grammar for the raw terms of the simply typed λ-calculus:

$$\begin{array}{lll} \alpha & ::= & c \mid \alpha \to \alpha \\ \Sigma & ::= & \cdot \mid \Sigma,[c : \mathrm{type}] \mid \Sigma,[c{:}\alpha] \\ \Gamma & ::= & \cdot \mid \Gamma,[x{:}\alpha] \\ \mathrm{A} & ::= & c \mid X \mid \mathrm{A}^1\ \mathrm{A}^2 \mid \lambda X_\alpha.\mathrm{A} \end{array}$$

- **Then:** Define all the operations that are possible at the “raw terms level”, e.g. realize that signatures and contexts are partial functions to types.

Simply Typed λ-Calculus as an Inference System: Judgments

- **Definition 3.49.** Judgments make statements about complex properties of the syntactic entities defined by the grammar.
- **Definition 3.50.** Judgments for the simply typed λ-calculus

$\vdash \Sigma : \mathsf{sig}$	Σ is a well-formed signature
$\Sigma \vdash \alpha : \mathsf{type}$	α is a well-formed type given the type assumptions in Σ
$\Sigma \vdash \Gamma : \mathsf{ctx}$	Γ is a well-formed context given the type assumptions in Σ
$\Gamma \vdash_{\Sigma} \mathrm{A} : \alpha$	A has type α given the type assumptions in Σ and Γ

Simply Typed λ-Calculus as an Inference System: Rules

- $A \in wff_\alpha(\Sigma_\mathcal{T}, \mathcal{V}_\mathcal{T})$, iff $\Gamma \vdash_\Sigma A : \alpha$ derivable in

$$\frac{\Sigma \vdash \Gamma : \mathrm{ctx} \quad \Gamma(X) = \alpha}{\Gamma \vdash_\Sigma X : \alpha}\,\mathit{wff\ var} \qquad \frac{\Sigma \vdash \Gamma : \mathrm{ctx} \quad \Sigma(c) = \alpha}{\Gamma \vdash_\Sigma c : \alpha}\,\mathit{wff\ const}$$

$$\frac{\Gamma \vdash_\Sigma A : \beta \to \alpha \quad \Gamma \vdash_\Sigma B : \beta}{\Gamma \vdash_\Sigma A\ B : \alpha}\,\mathit{wff\ app} \qquad \frac{\Gamma, [X{:}\beta] \vdash_\Sigma A : \alpha}{\Gamma \vdash_\Sigma \lambda X_\beta . A : \beta \to \alpha}\,\mathit{wff\ abs}$$

- **Oops:** this looks surprisingly like a natural deduction calculus. (⇝ Curry Howard Isomorphism)
- To be complete, we need rules for well-formed signatures, types and contexts

$$\frac{}{\vdash \cdot : \mathrm{sig}}\,\mathit{sig\ empty} \qquad \frac{\vdash \Sigma : \mathrm{sig}}{\vdash (\Sigma, [\alpha : \mathrm{type}]) : \mathrm{sig}}\,\mathit{sig\ type}$$

$$\frac{\vdash \Sigma : \mathrm{sig} \quad \Sigma \vdash \alpha : \mathrm{type}}{\vdash (\Sigma, [c{:}\alpha]) : \mathrm{sig}}\,\mathit{sig\ const}$$

$$\frac{\Sigma \vdash \alpha : \mathrm{type} \quad \Sigma \vdash \beta : \mathrm{type}}{\Sigma \vdash (\alpha \to \beta) : \mathrm{type}}\,\mathit{typ\ fn} \qquad \frac{\vdash \Sigma : \mathrm{sig} \quad \Sigma(\alpha) = \mathrm{type}}{\Sigma \vdash \alpha : \mathrm{type}}\,\mathit{typ\ start}$$

$$\frac{\vdash \Sigma : \mathrm{sig}}{\Sigma \vdash \cdot : \mathrm{ctx}}\,\mathit{ctx\ empty} \qquad \frac{\Sigma \vdash \Gamma : \mathrm{ctx} \quad \Sigma \vdash \alpha : \mathrm{type}}{\Sigma \vdash (\Gamma, [X{:}\alpha]) : \mathrm{ctx}}\,\mathit{ctx\ var}$$

Example: A Well-Formed Signature

- Let $\Sigma{:=}[\alpha:\mathsf{type}],[f{:}\alpha\to\alpha\to\alpha]$, then Σ is a well-formed signature, since we have derivations $\mathcal{A}$ and $\mathcal{B}$

$$\frac{\vdash\cdot:\mathsf{sig}}{\vdash[\alpha:\mathsf{type}]:\mathsf{sig}}\,\mathit{sig\ type}\qquad\frac{\mathcal{A}\quad[\alpha:\mathsf{type}](\alpha)=\mathsf{type}}{[\alpha:\mathsf{type}]\vdash\alpha:\mathsf{type}}\,\mathit{typ\ start}$$

and with these we can construct the derivation $\mathcal{C}$

$$\frac{\mathcal{A}\quad\dfrac{\mathcal{B}\quad\dfrac{\mathcal{B}\quad\mathcal{B}}{[\alpha:\mathsf{type}]\vdash(\alpha\to\alpha):\mathsf{type}}\,\mathit{typ\ fn}}{[\alpha:\mathsf{type}]\vdash(\alpha\to\alpha\to\alpha):\mathsf{type}}\,\mathit{typ\ fn}}{\vdash\Sigma:\mathsf{sig}}\,\mathit{sig\ const}$$

Example: A Well-Formed λ-Term

- using Σ from above, we can show that $\Gamma:=[X:\alpha]$ is a well-formed context:

$$\dfrac{\dfrac{\mathcal{C}}{\Sigma \vdash \cdot : \mathrm{ctx}}\,\textit{ctx empty} \quad \dfrac{\mathcal{C} \quad \Sigma(\alpha) = \mathrm{type}}{\Sigma \vdash \alpha : \mathrm{type}}\,\textit{typ start}}{\Sigma \vdash \Gamma : \mathrm{ctx}}\,\textit{ctx var}$$

  We call this derivation $\mathcal{G}$ and use it to show that

- $\lambda X_\alpha.f\;X\;X$ is well-typed and has type $\alpha \to \alpha$ in Σ. This is witnessed by the type derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{C} \quad \Sigma(f) = \alpha \to \alpha \to \alpha}{\Gamma \vdash_\Sigma f : \alpha \to \alpha \to \alpha}\,\textit{wff const} \quad \dfrac{\mathcal{G}}{\Gamma \vdash_\Sigma X : \alpha}\,\textit{wff var}}{\Gamma \vdash_\Sigma f\;X : \alpha \to \alpha}\,\textit{wff app} \quad \dfrac{\mathcal{G}}{\Gamma \vdash_\Sigma X : \alpha}\,\textit{wff var}}{\Gamma \vdash_\Sigma f\;X\;X : \alpha}\,\textit{wff app}}{\cdot \vdash_\Sigma \lambda X_\alpha.f\;X\;X : \alpha \to \alpha}\,\textit{wff abs}$$

$\beta\eta$-Equality by Inference Rules: One-Step Reduction

- One-step Reduction $(+\in\{\alpha,\beta,\eta\})$

$$\frac{\Gamma,[X:\beta]\vdash_\Sigma A:\alpha \quad \Gamma\vdash_\Sigma B:\beta}{\Gamma\vdash_\Sigma (\lambda X.A)\;B\rightarrow^1_\beta[B/X](A)}\;\mathit{wff}\,\beta\;\mathit{top}$$

$$\frac{\Gamma\vdash_\Sigma A:\beta\rightarrow\alpha \quad X\notin\mathrm{dom}(\Gamma)}{\Gamma\vdash_\Sigma \lambda X.A\;X\rightarrow^1_\eta A}\;\mathit{wff}\,\eta\;\mathit{top}$$

$$\frac{\Gamma\vdash_\Sigma A\rightarrow^1_+ B \quad \Gamma\vdash_\Sigma A\;C:\alpha}{\Gamma\vdash_\Sigma A\;C\rightarrow^1_+ B\;C}\;\mathit{tr\;appfn}$$

$$\frac{\Gamma\vdash_\Sigma A\rightarrow^1_+ B \quad \Gamma\vdash_\Sigma C\;A:\alpha}{\Gamma\vdash_\Sigma C\;A\rightarrow^1_+ C\;B}\;\mathit{tr\;apparg}$$

$$\frac{\Gamma,[X:\alpha]\vdash_\Sigma A\rightarrow^1_+ B}{\Gamma\vdash_\Sigma \lambda X.A\rightarrow^1_+ \lambda X.B}\;\mathit{tr\;abs}$$

$\beta\eta$-Equality by Inference Rules: Multi-Step Reduction

- Multi-Step-Reduction ($+\in\{\alpha,\beta,\eta\}$)

$$\frac{\Gamma \vdash_\Sigma A \rightarrow^1_+ B}{\Gamma \vdash_\Sigma A \rightarrow^*_+ B}\,\textit{ms start} \qquad \frac{\Gamma \vdash_\Sigma A : \alpha}{\Gamma \vdash_\Sigma A \rightarrow^*_+ A}\,\textit{ms ref}$$

$$\frac{\Gamma \vdash_\Sigma A \rightarrow^*_+ B \quad \Gamma \vdash_\Sigma B \rightarrow^*_+ C}{\Gamma \vdash_\Sigma A \rightarrow^*_+ C}\,\textit{ms trans}$$

- Congruence Relation

$$\frac{\Gamma \vdash_\Sigma A \rightarrow^*_+ B}{\Gamma \vdash_\Sigma A =_+ B}\,\textit{eq start}$$

$$\frac{\Gamma \vdash_\Sigma A =_+ B}{\Gamma \vdash_\Sigma B =_+ A}\,\textit{eq sym} \qquad \frac{\Gamma \vdash_\Sigma A =_+ B \quad \Gamma \vdash_\Sigma B =_+ C}{\Gamma \vdash_\Sigma A =_+ C}\,\textit{eq trans}$$

Type Computation: Manage Types Algorithmically

- **Questions:**

  type check: Is $\Gamma \vdash_{\Sigma} A \colon \alpha$?

  type inference: are there Γ, α, such that $\Gamma \vdash_{\Sigma} A \colon \alpha$?

  type reconstruction the above without type annotations at bound variables?
- prenex fragment makes problems decidable (see Curry Howard)
- **Algorithm (Hindley & Milner):**
  - invert inference rules
  - first-order unification,
  - universal generalization, minimization

Example Computation

rule tree

$$\dfrac{\dfrac{\dfrac{[X{:}\alpha]}{\Gamma,[X{:}\beta]}}{\Gamma,[X{:}\beta]\vdash_{\Sigma}X:\alpha}\qquad \Gamma\vdash_{\Sigma}\lambda X.X:\beta\to\alpha}{\dfrac{\dfrac{[W{:}\delta]\in\Gamma,[Z{:}\gamma]}{\Gamma,[Z{:}\gamma]\vdash_{\Sigma}W:\delta}}{\Gamma\vdash_{\Sigma}\lambda Z.W:\beta}}\ \Gamma\vdash_{\Sigma}\lambda X.X(\lambda Z.W):\alpha$$

constraint

$\alpha=\beta$,
$[W{:}\delta]\in\Gamma$,
$\beta=\gamma\to\delta$

- unification: $\alpha=\beta=\gamma\to\delta$,
- minimization: $\Gamma=[W{:}\delta]$
- **Solution:** $[W{:}\delta]]\vdash_{\Sigma}\lambda X.X(\lambda Z.W):\forall\gamma.\gamma\to\delta$

1.3.5 The Semantics of the Simply Typed λ-Calculus

Semantics of $\Lambda^{\rightarrow}$

- **Definition 3.51.** We call a collection $\mathcal{D}_{\mathcal{T}}{:=}\{\mathcal{D}_\alpha|\alpha{\in}\mathcal{T}\}$ a typed collection (of sets) and a collection $f_{\mathcal{T}}\colon \mathcal{D}_{\mathcal{T}}{\to}\mathcal{E}_{\mathcal{T}}$, a typed function, iff $f_\alpha\colon \mathcal{D}_\alpha{\to}\mathcal{E}_\alpha$.
- **Definition 3.52.** A typed collection $\mathcal{D}_{\mathcal{T}}$ is called a frame, iff $\mathcal{D}_{(\alpha\to\beta)} \subseteq \mathcal{D}_\alpha \to \mathcal{D}_\beta$
- **Definition 3.53.** Given a frame $\mathcal{D}_{\mathcal{T}}$, and a typed function $\mathcal{I}\colon \Sigma{\to}\mathcal{D}$, then we call $\mathcal{I}_\varphi\colon \mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}}){\to}\mathcal{D}$ the value function induced by $\mathcal{I}$, iff
  - $\mathcal{I}_\varphi|_{\mathcal{V}_{\mathcal{T}}} = \varphi, \qquad \mathcal{I}_\varphi|_{\Sigma_{\mathcal{T}}} = \mathcal{I}$
  - $\mathcal{I}_\varphi(\mathrm{A}\,\mathrm{B}) = \mathcal{I}_\varphi(\mathrm{A})(\mathcal{I}_\varphi(\mathrm{B}))$
  - $\mathcal{I}_\varphi(\lambda X_\alpha.\mathrm{A})$ is that function $f{\in}\mathcal{D}_{(\alpha\to\beta)}$, such that $f(a) = \mathcal{I}_{\varphi,[a/X]}(\mathrm{A})$ for all $a{\in}\mathcal{D}_\alpha$
- **Definition 3.54.** We call a frame $\langle\mathcal{D},\mathcal{I}\rangle$ comprehension closed or a $\Sigma_{\mathcal{T}}$-algebra, iff $\mathcal{I}_\varphi\colon \mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}}){\to}\mathcal{D}$ is total. (every λ-term has a value)

1.3.5.1 Soundness of the Simply Typed λ-Calculus

Substitution Value Lemma for λ-Terms I

- **Lemma 3.55 (Substitution Value Lemma).** *Let* A *and* B *be terms, then* $\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A})) = \mathcal{I}_\psi(\mathrm{A})$, *where* $\psi = \varphi,[\mathcal{I}_\varphi(\mathrm{B})/X]$
- *Proof:* by induction on the depth of A

  we have five cases

  1. $\mathrm{A} = X$
     1.1. Then $\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A})) = \mathcal{I}_\varphi([\mathrm{B}/X](X)) = \mathcal{I}_\varphi(\mathrm{B}) = \psi(X) = \mathcal{I}_\psi(X) = \mathcal{I}_\psi(\mathrm{A})$.
  2. $\mathrm{A} = Y \neq X$ and $Y \in \mathcal{V}_\mathcal{T}$
     2.1. then $\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A})) = \mathcal{I}_\varphi([\mathrm{B}/X](Y)) = \mathcal{I}_\varphi(Y) = \varphi(Y) = \psi(Y) = \mathcal{I}_\psi(Y) = \mathcal{I}_\psi(\mathrm{A})$.
  3. $\mathrm{A} \in \Sigma_\mathcal{T}$
     3.1. This is analogous to the last case.
  4. $\mathrm{A} = \mathrm{C}\ \mathrm{D}$
     4.1. then $\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A})) = \mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{C}\ \mathrm{D})) = \mathcal{I}_\varphi(([\mathrm{B}/X](\mathrm{C}))\ ([\mathrm{B}/X](\mathrm{D}))) = \mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{C}))(\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{D}))) = \mathcal{I}_\psi(\mathrm{C})(\mathcal{I}_\psi(\mathrm{D})) = \mathcal{I}_\psi(\mathrm{C}\ \mathrm{D}) = \mathcal{I}_\psi(\mathrm{A})$

Substitution Value Lemma for λ-Terms II

5. $\mathrm{A} = \lambda Y_\alpha . \mathrm{C}$
   5.1. We can assume that $X \neq Y$ and $Y \notin \mathrm{free}(\mathrm{B})$
   5.2. Thus for all $a \in \mathcal{D}_\alpha$ we have
   $\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A}))(a) = \mathcal{I}_\varphi([\mathrm{B}/X](\lambda Y.\mathrm{C}))(a) = \mathcal{I}_\varphi(\lambda Y.[\mathrm{B}/X](\mathrm{C}))(a) = \mathcal{I}_{\varphi,[a/Y]}([\mathrm{B}/X](\mathrm{C})) = \mathcal{I}_{\psi,[a/Y]}(\mathrm{C}) = \mathcal{I}_\psi(\lambda Y.\mathrm{C})(a) = \mathcal{I}_\psi(\mathrm{A})(a)$

Soundness of $\alpha\beta\eta$-Equality

- **Theorem 3.56.** *Let* $\mathcal{A}:=\langle\mathcal{D},\mathcal{I}\rangle$ *be a* $\Sigma_{\mathcal{T}}$*-algebra and* $Y\notin free(\mathrm{A})$*, then* $\mathcal{I}_\varphi(\lambda X.\mathrm{A})=\mathcal{I}_\varphi(\lambda Y.[Y/X]\mathrm{A})$ *for all assignments* φ*.*
- *Proof:* by substitution value lemma

$$\begin{aligned}\mathcal{I}_\varphi(\lambda Y.[Y/X]\mathrm{A})@\mathsf{a} &= \mathcal{I}_{\varphi,[\mathsf{a}/Y]}([Y/X](\mathrm{A}))\\ &= \mathcal{I}_{\varphi,[\mathsf{a}/X]}(\mathrm{A})\\ &= \mathcal{I}_\varphi(\lambda X.\mathrm{A})@\mathsf{a}\end{aligned}$$

- **Theorem 3.57.** *If* $\mathcal{A}:=\langle\mathcal{D},\mathcal{I}\rangle$ *is a* $\Sigma_{\mathcal{T}}$*-algebra and* X *not bound in* A*, then* $\mathcal{I}_\varphi((\lambda X.\mathrm{A})\ \mathrm{B})=\mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A}))$*.*

  Proof: by substitution value lemma again
-

$$\begin{aligned}\mathcal{I}_\varphi((\lambda X.\mathrm{A})\ \mathrm{B}) &= \mathcal{I}_\varphi(\lambda X.\mathrm{A})@\mathcal{I}_\varphi(\mathrm{B})\\ &= \mathcal{I}_{\varphi,[\mathcal{I}_\varphi(\mathrm{B})/X]}(\mathrm{A})\\ &= \mathcal{I}_\varphi([\mathrm{B}/X](\mathrm{A}))\end{aligned}$$

Soundness of $\alpha\beta\eta$ (continued)

- **Theorem 3.58.** *If* $X \notin \mathrm{free}(\mathrm{A})$, *then* $\mathcal{I}_\varphi(\lambda X.\mathrm{A}\ X) = \mathcal{I}_\varphi(\mathrm{A})$ *for all* φ*.*
- *Proof:* by calculation

$$\begin{aligned}
\mathcal{I}_\varphi(\lambda X.\mathrm{A}\ X)@\mathsf{a} &= \mathcal{I}_{\varphi,[\mathsf{a}/X]}(\mathrm{A}\ X) \\
&= \mathcal{I}_{\varphi,[\mathsf{a}/X]}(\mathrm{A})@\mathcal{I}_{\varphi,[\mathsf{a}/X]}(X) \\
&= \mathcal{I}_\varphi(\mathrm{A})@\mathcal{I}_{\varphi,[\mathsf{a}/X]}(X) \qquad \text{as } X \notin \mathrm{free}(\mathrm{A}). \\
&= \mathcal{I}_\varphi(\mathrm{A})@\mathsf{a}
\end{aligned}$$

- **Theorem 3.59.** *$\alpha\beta\eta$-equality is sound wrt. $\Sigma_{\mathcal{T}}$-algebras. (if* $\mathrm{A} =_{\alpha\beta\eta} \mathrm{B}$, *then* $\mathcal{I}_\varphi(\mathrm{A}) = \mathcal{I}_\varphi(\mathrm{B})$ *for all assignments* φ*)*

1.3.5.2 Completeness of $\alpha\beta\eta$-Equality

Normal Forms in the simply typed λ-calculus

- **Definition 3.60.** We call a term $\mathrm{A}\in wff_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$ a β normal form iff there is no $\mathrm{B}\in wff_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$ with $\mathrm{A}\rightarrow_{\beta}\mathrm{B}$.
  We call N a β normal form of A, iff N is a β-normal form and $\mathrm{A}\rightarrow_{\beta}\mathrm{N}$.
  We denote the set of β-normal forms with $wff_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})\big\downarrow_{\beta\eta}$.
- We have just proved that $\beta\eta$-reduction is terminating and confluent, so we have
- **Corollary 3.61 (Normal Forms).** *Every* $\mathrm{A}\in wff_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$ *has a unique* β *normal form (*$\beta\eta$*, long* $\beta\eta$ *normal form), which we denote by* $\mathrm{A}\downarrow_{\beta}$ *(*$\mathrm{A}\downarrow_{\beta\eta}$ $\mathrm{A}\downarrow_{\beta\eta'}$*)*

Frames and Quotients

- **Definition 3.62.** Let $\mathcal{D}$ be a frame and $\sim$ a typed equivalence relation on $\mathcal{D}$, then we call $\sim$ a congruence on $\mathcal{D}$, iff $f \sim f'$ and $g \sim g'$ imply $f(g) \sim f'(g')$.
- **Definition 3.63.** We call a congruence $\sim$ functional, iff for all $f, g \in \mathcal{D}_{(\alpha\to\beta)}$ the fact that $f(a) \sim g(a)$ holds for all $a \in \mathcal{D}_\alpha$ implies that $f \sim g$.
- **Example 3.64.** $=_\beta$ ($=_{\beta\eta}$) is a (functional) congruence on $cwff_{\mathcal{T}}(\Sigma_{\mathcal{T}})$ by definition.
- **Theorem 3.65.** *Let $\mathcal{D}\mathcal{T}$ be a $\Sigma_{\mathcal{T}}$-frame and $\sim$ a functional congruence on $\mathcal{D}$, then the quotient space $\mathcal{D}/\sim$ is a $\Sigma_{\mathcal{T}}$-frame.*
- *Proof:*
  1. $\mathcal{D}/\sim = \{f_\sim | f \in \mathcal{D}\}$, define $f_\sim(a_\sim) := f(a)_\sim$.
  2. This only depends on equivalence classes: Let $f' \in f_\sim$ and $a' \in a_\sim$.
  3. Then $f(a)_\sim = f'(a)_\sim = f'(a')_\sim = f(a')_\sim$
  4. To see that we have $f_\sim = g_\sim$, iff $f \sim g$, iff $f(a) = g(a)$ since $\sim$ is functional.
  5. This is the case iff $f(a)_\sim = g(a)_\sim$, iff $f_\sim(a_\sim) = g_\sim(a_\sim)$ for all $a \in \mathcal{D}_\alpha$ and thus for all $a_\sim \in \mathcal{D}/\sim$.

$\beta\eta$-Equivalence as a Functional Congruence

- **Lemma 3.66.** *$\beta\eta$-equality is a functional congruence on $\mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$.*
- *Proof:* Let $\mathrm{A}\ \mathrm{C}=_{\beta\eta}\mathrm{B}\ \mathrm{C}$ for all C and $X\in(\mathcal{V}_\gamma\backslash\mathrm{free}(\mathrm{A})\cup\mathrm{free}(\mathrm{B}))$.
  1. then (in particular) $\mathrm{A}\ X=_{\beta\eta}\mathrm{B}\ X$, and
  2. $(\lambda X.\mathrm{A}\ X)=_{\beta\eta}(\lambda X.\mathrm{B}\ X)$, since $\beta\eta$-equality acts on subterms.
  3. By definition we have $\mathrm{A}=_\eta(\lambda X_\alpha.\mathrm{A}\ X)=_{\beta\eta}(\lambda X_\alpha.\mathrm{B}\ X)=_\eta\mathrm{B}$.
- **Definition 3.67.** We call an injective substitution $\sigma\colon\mathrm{free}(\mathrm{C})\to\Sigma_{\mathcal{T}}$ a grounding substitution for $\mathrm{C}\in\mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$, iff no $\sigma(X)$ occurs in C.
- **Observation:** They always exist, since all Σ_α are infinite and free(C) is finite.
- **Theorem 3.68.** *$\beta\eta$-equality is a functional congruence on $\mathit{cwff}_{\mathcal{T}}(\Sigma_{\mathcal{T}})$.*
- *Proof:* We use **??**
  1. Let $\mathrm{A},\mathrm{B}\in\mathit{cwff}_{(\alpha\to\beta)}(\Sigma_{\mathcal{T}})$, such that $\mathrm{A}\neq_{\beta\eta}\mathrm{B}$.
  2. As $\beta\eta$ is functional on $\mathit{wff}_{\mathcal{T}}(\Sigma_{\mathcal{T}},\mathcal{V}_{\mathcal{T}})$, there must be a C with $\mathrm{A}\ \mathrm{C}\neq_{\beta\eta}\mathrm{B}\ \mathrm{C}$.
  3. Now let $\mathrm{C}':=\sigma(\mathrm{C})$, for a grounding substitution σ.
  4. Any $\beta\eta$ conversion sequence for $\mathrm{A}\ \mathrm{C}'\neq_{\beta\eta}\mathrm{B}\ \mathrm{C}'$ induces one for $\mathrm{A}\ \mathrm{C}\neq_{\beta\eta}\mathrm{B}\ \mathrm{C}$.
  5. Thus we have shown that $\mathrm{A}\neq_{\beta\eta}\mathrm{B}$ entails $\mathrm{A}\ \mathrm{C}'\neq_{\beta\eta}\mathrm{B}\ \mathrm{C}'$.

A Herbrand Model for $\Lambda^{\rightarrow}$

- **Definition 3.69.** We call $\mathcal{T}_{\beta\eta}:=\langle \mathit{cwff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}){\downarrow}_{\beta\eta}, \mathcal{I}^{\beta\eta}\rangle$ the Σ term algebra, if $\mathcal{I}^{\beta\eta} = \mathsf{Id}_{\Sigma_{\mathcal{T}}}$.
- The name "term algebra" in the previous definition is justified by the following
- **Theorem 3.70.** *$\mathcal{T}_{\beta\eta}$ is a $\Sigma_{\mathcal{T}}$-algebra*
- *Proof:* We use the work we did above
  1. Note that $\mathit{cwff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}){\downarrow}_{\beta\eta} = \mathit{cwff}_{\mathcal{T}}(\Sigma_{\mathcal{T}})/{=_{\beta\eta}}$ and thus a $\Sigma_{\mathcal{T}}$-frame by **??** and **??**.
  2. So we only have to show that the value function $\mathcal{I}^{\beta\eta} = \mathsf{Id}_{\Sigma_{\mathcal{T}}}$ is total.
  3. Let φ be an assignment into $\mathit{cwff}_{\mathcal{T}}(\Sigma_{\mathcal{T}}){\downarrow}_{\beta\eta}$.
  4. Note that $\sigma := \varphi|_{\mathsf{free}(\mathsf{A})}$ is a substitution, since $\mathsf{free}(\mathsf{A})$ is finite.
  5. A simple induction on the structure of A shows that $\mathcal{I}^{\beta\eta}_{\varphi}(\mathsf{A}) = (\sigma(\mathsf{A})){\downarrow}_{\beta\eta}$.
  6. So the value function is total since substitution application is.

Completetness of $\alpha\beta\eta$-Equality

- **Theorem 3.71.** $A = B$ *is valid in the class of* $\Sigma_{\mathcal{T}}$*-algebras, iff* $A =_{\alpha\beta\eta} B$*.*
- *Proof:* For A, B closed this is a simple consequence of the fact that $\mathcal{T}_{\beta\eta}$ is a $\Sigma_{\mathcal{T}}$-algebra.
  1. If $A = B$ is valid in all $\Sigma_{\mathcal{T}}$-algebras, it must be in $\mathcal{T}_{\beta\eta}$ and in particular $A{\downarrow}_{\beta\eta} = \mathcal{I}^{\beta\eta}(A) = \mathcal{I}^{\beta\eta}(B) = B{\downarrow}_{\beta\eta}$ and therefore $A =_{\alpha\beta\eta} B$.

  If the equation has free variables, then the argument is more subtle.

  2. Let σ be a grounding substitution for A and B and φ the induced variable assignment.
  3. Thus $\mathcal{I}^{\beta\eta}{}_{\varphi}(A) = \mathcal{I}^{\beta\eta}{}_{\varphi}(B)$ is the $\beta\eta$-normal form of $\sigma(A)$ and $\sigma(B)$.
  4. Since φ is a structure preserving homomorphism on well-formed formulae, $\varphi^{-1}(\mathcal{I}^{\beta\eta}{}_{\varphi}(A))$ is the is the $\beta\eta$-normal form of both A and B and thus $A =_{\alpha\beta\eta} B$.

1.3.6 De Bruijn Indices

De Bruijn Indices: Nameless Dummies for Bound Variables

- **Problem:** We consider alphabetically equal λ terms as "syntactically equal".
- **Idea:** Get rid of variables by replacing them with nameless dummies (numbers).
- **Definition 3.72 (Formally).**
  Raw λ-terms with de Bruijn indices are expressions given by changing the last production in 3.48 to

  $$\mathrm{A} \quad ::= \quad c \mid n \mid \mathrm{A}^1\ \mathrm{A}^2 \mid \lambda\mathrm{A}$$

  A variable n is bound if it is in the scope of at least n binders (λ); otherwise it is free. The binding site for a variable n is the nth binder it is in the scope of, starting from the innermost binder.
- **Example 3.73.** $(\lambda x.\lambda y.z\ x\ (\lambda u.u\ x))\ (\lambda w.w\ x)$, becomes $(\lambda\lambda 4\ 2\ (\lambda 1\ 3))\ (\lambda 5\ 1)$,
- **Problem:** De Bruijn indices are less readable than standard λ terms.
- **Solution:** Maintain a UI with names even when using de Bruijn indices internally.
- **Problem:** Substitution and β reduction become complicated. (see below)

De Bruijn Indices: β-Reduction

- **Definition 3.74.** For β-reducing $(\lambda \mathrm{M})\ \mathrm{N}$ we must:
  1. find variable occurrences $n_1, n_2, \ldots, n_k$ in M bound by outer λ in $\lambda\mathrm{M}$

- **Example 3.75.** We perform the steps outlined above on $(\lambda\lambda 4\ 2\ (\lambda 1\ 3))\ (\lambda 5\ 1)$:
  1. we obtain $\lambda 4\ n_1\ (\lambda 1\ n_2)$

De Bruijn Indices: β-Reduction

- **Definition 3.76.** For β-reducing $(\lambda \mathrm{M})\ \mathrm{N}$ we must:
  1. find variable occurrences $n_1, n_2, \ldots, n_k$ in M bound by outer λ in $\lambda\mathrm{M}$
  2. decrement the free variables of M to match the removal of the outer λ,

- **Example 3.77.** We perform the steps outlined above on $(\lambda\lambda 4\ 2\ (\lambda 1\ 3))\ (\lambda 5\ 1)$:
  1. we obtain $\lambda 4\ n_1\ (\lambda 1\ n_2)$
  2. we obtain $\lambda 3\ n_1\ (\lambda 1\ n_2)$ decrementing free variables.

De Bruijn Indices: β-Reduction

- **Definition 3.78.** For β-reducing $(\lambda \mathrm{M})\ \mathrm{N}$ we must:
  1. find variable occurrences $n_1, n_2, \ldots, n_k$ in M bound by outer λ in λM
  2. decrement the free variables of M to match the removal of the outer λ,
  3. replace n_i with N, suitably incrementing the free variables in N each time, to match the number of λ-binders, under which n_i occurs.
- **Example 3.79.** We perform the steps outlined above on $(\lambda\lambda 4\ 2\ (\lambda 1\ 3))\ (\lambda 5\ 1)$:
  1. we obtain $\lambda 4\ n_1\ (\lambda 1\ n_2)$
  2. we obtain $\lambda 3\ n_1\ (\lambda 1\ n_2)$ decrementing free variables.
  3. we replace X with the argument $\lambda 5\ 1$.
     - n_1 is under one $\lambda \rightsquigarrow$ replace it with $\lambda 6\ 1$
     - n_2 is under two λs $\rightsquigarrow$ replace it with $\lambda 7\ 1$.

  The final result is $\lambda 3\ (\lambda 6\ 1)\ (\lambda 1\ (\lambda 7\ 1))$

1.3.7 Simple Type Theory

Higher-Order Logic Revisited

- **Idea:** introduce special base type prop for truth values
- **Definition 3.80.** We call a Σ-algebra $\langle\mathcal{D},\mathcal{I}\rangle$ a Henkin model, iff $\mathcal{D}_{\mathrm{prop}} = \{\mathsf{T},\mathsf{F}\}$.
- **Definition 3.81.** $\mathrm{A}_{\mathrm{prop}}$ valid under φ, iff $\mathcal{I}_\varphi(\mathrm{A}) = \mathsf{T}$
- **Definition 3.82.** Connectives in Σ: $\neg\in\Sigma_{\mathrm{prop}\to\mathrm{prop}}$ and $\{\vee,\wedge,\Rightarrow,\Leftrightarrow,\ldots\}\subseteq\Sigma_{\mathrm{prop}\to\mathrm{prop}\to\mathrm{prop}}$ (with the intuitive $\mathcal{I}$-values)
- **Definition 3.83.** Quantifiers: $\Pi^\alpha\in\Sigma_{\alpha\to\mathrm{prop}\to\mathrm{prop}}$ with $\mathcal{I}(\Pi^\alpha)(p) = \mathsf{T}$, iff $p(a) = \mathsf{T}$ for all $a\in\mathcal{D}_\alpha$.
- **Definition 3.84.** [Quantified] formulae: $\forall X_\alpha.\mathrm{A}$ stands for $\Pi^\alpha\,(\lambda X_\alpha.\mathrm{A})$.
- $\mathcal{I}_\varphi(\forall X_\alpha.\mathrm{A}) = \mathcal{I}(\Pi^\alpha)(\mathcal{I}_\varphi(\lambda X_\alpha.\mathrm{A})) = \mathsf{T}$, iff $\mathcal{I}_{\varphi,[a/X]}(\mathrm{A}) = \mathsf{T}$ for all $a\in\mathcal{D}_\alpha$
- looks like PLΩ (Call any such system $\mathrm{HOL}^\to$)

Higher-Order Abstract Syntax

- **Idea:** In $\mathrm{HOL}^{\rightarrow}$, we already have variable binder: λ, use that to treat quantification.
- **Definition 3.85.** We assume logical constants Π^α and σ^α of type $\alpha \to \mathrm{prop} \to \mathrm{prop}$.
  Regain quantifiers as abbreviations:
  $$(\forall X_\alpha.A) := \Pi^\alpha\ (\lambda X_\alpha.A) \qquad (\exists X_\alpha.A) := \sigma^\alpha\ (\lambda X_\alpha.A)$$
- **Definition 3.86.** We must fix the semantics of logical constants:
  1. $\mathcal{I}(\Pi^\alpha)(p) = \mathsf{T}$, iff $p(a) = \mathsf{T}$ for all $a \in \mathcal{D}_\alpha$ (i.e. if p is the universal set)
  2. $\mathcal{I}(\sigma^\alpha)(p) = \mathsf{T}$, iff $p(a) = \mathsf{T}$ for some $a \in \mathcal{D}_\alpha$ (i.e. iff p is non-empty)
- With this, we re-obtain the semantics we have given for quantifiers above:

$$\mathcal{I}_\varphi(\forall X_\iota.A) = \mathcal{I}_\varphi(\Pi^\iota\ (\lambda X_\iota.A)) = \mathcal{I}(\Pi^\iota)(\mathcal{I}_\varphi(\lambda X_\iota.A)) = \mathsf{T}$$

iff $\mathcal{I}_\varphi(\lambda X_\iota.A)(a) = \mathcal{I}_{[a/X],\varphi}(A) = \mathsf{T}$ for all $a \in \mathcal{D}_\alpha$

Alternative: HOL$^{\infty}$

- only one logical constant $q^{\alpha}\in\Sigma_{\alpha\to\alpha\to\mathrm{prop}}$ with $\mathcal{I}(q^{\alpha})(a,b)=\mathrm{T}$, iff $a=b$.
- Definitions (D) and Notations (N)

N	$\mathrm{A}_\alpha = \mathrm{B}_\alpha$	for	$q^{\alpha}\mathrm{A}_\alpha \mathrm{B}_\alpha$
D	T	for	$q^{\mathrm{prop}} = q^{\mathrm{prop}}$
D	F	for	$\lambda X_{\mathrm{prop}}.T = \lambda X_{\mathrm{prop}}.X_{\mathrm{prop}}$
D	Π^{α}	for	$q^{\alpha\to\mathrm{prop}}\ (\lambda X_\alpha.T)$
N	$\forall X_\alpha.\mathrm{A}$	for	$\Pi^{\alpha}\ (\lambda X_\alpha.\mathrm{A})$
D	$\wedge$	for	$\lambda X_{\mathrm{prop}}.\lambda Y_{\mathrm{prop}}.(\lambda G_{\mathrm{prop}\to\mathrm{prop}\to\mathrm{prop}}.G\,T\,T = \lambda G_{\mathrm{prop}\to\mathrm{prop}\to\mathrm{prop}}$
N	$\mathrm{A}\wedge\mathrm{B}$	for	$\wedge\ (\mathrm{A}_{\mathrm{prop}})\ (\mathrm{B}_{\mathrm{prop}})$
D	$\Rightarrow$	for	$\lambda X_{\mathrm{prop}}.\lambda Y_{\mathrm{prop}}.(X = X\wedge Y)$
N	$\mathrm{A}\Rightarrow\mathrm{B}$	for	$\Rightarrow\ (\mathrm{A}_{\mathrm{prop}})\ (\mathrm{B}_{\mathrm{prop}})$
D	$\neg$	for	$q^{\mathrm{prop}}\ F$
D	$\vee$	for	$\lambda X_{\mathrm{prop}}.\lambda Y_{\mathrm{prop}}.\neg(\neg X\wedge\neg Y)$
N	$\mathrm{A}\vee\mathrm{B}$	for	$\vee\ (\mathrm{A}_{\mathrm{prop}})\ (\mathrm{B}_{\mathrm{prop}})$
D	$\exists X_\alpha.\mathrm{A}_{\mathrm{prop}}$	for	$\neg(\forall X_\alpha.\neg\mathrm{A})$
N	$\mathrm{A}_\alpha\neq\mathrm{B}_\alpha$	for	$\neg q^{\alpha}\ (\mathrm{A}_\alpha)\ (\mathrm{B}_\alpha)$

- yield the intuitive meanings for connectives and quantifiers.

Henkin's Theorem

- **Theorem 3.87 (Henkin's Theorem).** *Every $\mathcal{H}_\Omega$-consistent set of sentences has a model.*
- *Proof:*
  1. Let Φ be a $\mathcal{H}_\Omega$-consistent set of sentences.
  2. Extend Φ by adding sentences until Φ becoms a Hintikka set $\mathcal{H}$ with good closure properties.
  3. Build a term Σ-algebra as a typed universe and interpret *TWFfcl*prop in $\mathcal{D}_{\mathsf{prop}}$ by setting $\mathcal{I}_\varphi(\mathrm{A}) = \mathsf{T}$, iff $\mathrm{A} \in \mathcal{H}$.
- **Theorem 3.88 (Completeness Theorem for $\mathcal{H}_\Omega$).** *If $\Phi \models \mathrm{A}$, then $\Phi \vdash_{\mathcal{H}_\Omega} \mathrm{A}$.*

  Proof: We prove the result by playing with negations.
- 1. If A is valid in all models of Φ, then $\Phi \cup \{\neg \mathrm{A}\}$ has no model
  2. Thus $\Phi \cup \{\neg \mathrm{A}\}$ is inconsistent by (the contrapositive of) Henkins Theorem.
  3. So $\Phi \vdash_{\mathcal{H}_\Omega} \neg\neg \mathrm{A}$ by negation introduction and thus $\Phi \vdash_{\mathcal{H}_\Omega} \mathrm{A}$ by negation elimination.

Consequences of Henkin's Theorem

- **Theorem 3.89 (Compactness).** *If $\mathcal{H} \models \mathrm{A}$, then there is a finite $\mathcal{K} \subseteq \mathcal{H}$ with $\mathcal{K} \models \mathrm{A}$.*
- **Theorem 3.90 (Higher-Order Löwenheim/Skolem).** *If A is satisfiable, then there is a countable Henkin model $\mathcal{M}$ with $\mathcal{M} \models \mathrm{A}$.*
- **Corollary 3.91 (Skolem-Paradox).** *$\mathbb{R}$ is uncountable (by Cantor's theorem), but has a countable Henkin model.*
- **Problem:** Is there a contradiction?
- **Remark:** Look at the *exact* logical formulation of Cantor's theorem, what does that mean in terms of Henkin models!
- **Turns Out:** There is no contradiction in $\neg(\exists F : \mathbb{N} \to \mathbb{R}.F$ surjective$)$
  - The non-existence of surjective functions only entails a cardinality difference for standard models.
  - in Henkin models it only means that $\mathcal{D}_{(\alpha \to \beta)}$ contains no surjective functions.
- **Gödel Theorems:** There is no formal system that can distinguish between Henkin and standard models.

Are there Functions at all in Henkin Models?

- **In General:** All that can be written down! ($\Sigma_{\mathcal{T}}$-algebras are comprehension closed)
  - Otherwise $\mathcal{D}_\alpha$ could be empty.
  - $\mathcal{D}_{\mathsf{prop}} \neq \emptyset$, as $\mathcal{D}_{\mathsf{prop}} \supseteq \{\mathsf{T}, \mathsf{F}\}$ as $\mathcal{I}_\varphi(\forall X_{\mathsf{prop}}.X \vee \neg X) = \mathsf{T}$ and $\mathcal{I}_\varphi(\forall X_{\mathsf{prop}}.X \wedge \neg X) = \mathsf{F}$.
- **What functions we write down?:**
  - $\mathcal{D}_{(\alpha\to\alpha)} \neq \emptyset$, since $\mathcal{I}_\varphi(\lambda X_\alpha.X) \in \mathcal{D}_{(\alpha\to\alpha)}$.
  - $\mathcal{D}_{(\mathsf{prop}\to\iota)} = \emptyset$, iff $\mathcal{D}_\iota = \emptyset$. ($\lambda X_{\mathsf{prop}}.Y_\iota$ does not help)
- **In General:** $\mathcal{D}_{(\alpha\to\beta)} = \emptyset$, sometimes! (Curry-Howard-Iso.)
- **Lambda-Definable Functions:**
  - are always total (terminate on any input)
  - e.g. on the natural numbers: $+, \cdot, \hat{}$ but not $-, /, \sqrt{}$
- **Idea:** Guarantee that $\mathcal{D}_\alpha \neq \emptyset$ by a constant $c \in \Sigma_\alpha$.
- **Problem:** But what are good constants that give us mathematically relevant function universes? (up next)

More Operators and Axioms for $\mathrm{HOL}^{\rightarrow}$

- **Definition 3.92.** The unary conditional $\mathrm{w}^{\alpha}\in\Sigma_{\mathrm{prop}\to\alpha\to\alpha}$
  $\mathrm{w}\ (\mathrm{A}_{\mathrm{prop}})\mathrm{B}_{\alpha}$ means: "If A, then B".
- **Definition 3.93.** The binary conditional $\mathrm{if}^{\alpha}\in\Sigma_{\mathrm{prop}\to\alpha\to\alpha\to\alpha}$
  $\mathrm{if}\ (\mathrm{A}_{\mathrm{prop}})\ (\mathrm{B}_{\alpha})\ (\mathrm{C}_{\alpha})$ means: "if A, then B else C".
- **Definition 3.94.** The description operator $\iota^{\alpha}\in\Sigma_{\alpha\to\mathrm{prop}\to\alpha}$
  if P is a singleton set, then $\iota\ (\mathrm{P}_{\alpha\to\mathrm{prop}})$ is the (unique) element in P.
- **Definition 3.95.** The choice operator $\gamma^{\alpha}\in\Sigma_{\alpha\to\mathrm{prop}\to\alpha}$
  if P is non-empty, then $\gamma\ (\mathrm{P}_{\alpha\to\mathrm{prop}})$ is an arbitrary element from P.
- **Definition 3.96 (Axioms for these Operators).**
  - unary conditional: $\forall\varphi_{\mathrm{prop}}.\forall X_{\alpha}.\varphi\Rightarrow \mathrm{w}\ \varphi X = X$
  - binary conditional: $\forall\varphi_{\mathrm{prop}}.\forall X_{\alpha}, Y_{\alpha}, Z_{\alpha}.(\varphi\Rightarrow \mathrm{if}\ \varphi\ X\ Y = X)\wedge(\neg\varphi\Rightarrow \mathrm{if}\ \varphi\ Z\ X = X)$
  - description operator $\forall P_{\alpha\to\mathrm{prop}}.(\exists^{1}X_{\alpha}.PX)\Rightarrow(\forall Y_{\alpha}.PY\Rightarrow\iota\ P = Y)$
  - choice operator $\forall P_{\alpha\to\mathrm{prop}}.(\exists X_{\alpha}.PX)\Rightarrow(\forall Y_{\alpha}.PY\Rightarrow\gamma\ P = Y)$
- **Idea:** These operators ensure a much larger supply of functions in Henkin models.

More on the Description Operator

- ι ! is a weak form of the choice operator (only works on singleton sets)
- Alternative Axiom of Descriptions: $\forall X_\alpha.\iota^\alpha \; = \; X = X$.
  - use that $\mathcal{I}_{[\mathsf{a}/X]}(= \; X) = \{\mathsf{a}\}$
  - we only need this for base types $\neq$ prop
  - Define $\iota^{\mathsf{prop}} := = \; (\lambda X_{\mathsf{prop}}.X)$ or $\iota^{\mathsf{prop}} := (\lambda G_{\mathsf{prop} \to \mathsf{prop}}.G \; T)$ or $\iota^{\mathsf{prop}} := = = \; T$
  - $\iota^{(\alpha \to \beta)} := (\lambda H_{\alpha \to \beta \to \mathsf{prop}} X_\alpha.\iota^\beta \; (\lambda Z_\beta.(\exists F_{\alpha \to \beta}.H \; F \wedge F \; X = Z)))$

1.4 Category Theory

1.4.1 Introduction

Common Structure to Mathematical Objects

- **Example 4.1.** Let A, B, and C be sets, and $f\colon A{\to}B$ and $g\colon B{\to}C$ be functions. Then $g\circ f$ is a function and we have functions Id_A and Id_B with $\mathrm{Id}_A\circ f=f=f\circ\mathrm{Id}_B$.

Common Structure to Mathematical Objects

- **Example 4.5.** Let A, B, and C be sets, and $f\colon A{\rightarrow}B$ and $g\colon B{\rightarrow}C$ be functions. Then $g \circ f$ is a function and we have functions Id_A and Id_B with $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.
- **Example 4.6.** Let A, B, and C be topological spaces, and $f\colon A{\rightarrow}B$ and $g\colon B{\rightarrow}C$ be continuous functions. Then $g \circ f$, Id_A, and Id_B are continuous and $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.

Common Structure to Mathematical Objects

- **Example 4.9.** Let A, B, and C be sets, and $f\colon A \to B$ and $g\colon B \to C$ be functions. Then $g \circ f$ is a function and we have functions Id_A and Id_B with $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.
- **Example 4.10.** Let A, B, and C be topological spaces, and $f\colon A \to B$ and $g\colon B \to C$ be continuous functions. Then $g \circ f$, Id_A, and Id_B are continuous and $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.
- **Example 4.11.** Let A, B, and C be posets, and $f\colon A \to B$ and $g\colon B \to C$ be monotone functions. Then $g \circ f$, Id_A, and Id_B are monotone and $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.

Common Structure to Mathematical Objects

- **Example 4.13.** Let A, B, and C be sets, and $f\colon A{\to}B$ and $g\colon B{\to}C$ be functions. Then $g \circ f$ is a function and we have functions Id_A and Id_B with $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.
- **Example 4.14.** Let A, B, and C be topological spaces, and $f\colon A{\to}B$ and $g\colon B{\to}C$ be continuous functions. Then $g \circ f$, Id_A, and Id_B are continuous and $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.
- **Example 4.15.** Let A, B, and C be posets, and $f\colon A{\to}B$ and $g\colon B{\to}C$ be monotone functions. Then $g \circ f$, Id_A, and Id_B are monotone and $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.
- **Example 4.16.** Let A, B, and C be monoids, and $f\colon A{\to}B$ and $g\colon B{\to}C$ be monoid homomorphisms. Then $g \circ f$, Id_A, and Id_B are monoid homomorphisms and $\mathrm{Id}_A \circ f = f = f \circ \mathrm{Id}_B$.

Categories: The Definition

- **Definition 4.17.**
  A category $\mathcal{C}$ consists of:
  1. A class $\mathrm{ob}(\mathcal{C})$ of objects.
  2. A class $\mathrm{Mor}_{\mathcal{C}}$ of arrows (also called morphism or map).
  3. For each arrow f, two objects which are called domain $\mathrm{dom}(f)$ and codomain $\mathrm{cod}(f)$ of f. We write $f\colon \mathrm{dom}(f) \to \mathrm{cod}(f)$ and call two arrows f and g composable, iff $\mathrm{dom}(f) = \mathrm{cod}(g)$.
  4. An associative operation $\circ$ called composition assigning to each pair (f,g) of composable arrows another arrow $g \circ f$ such that $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$ and $\mathrm{cod}(g \circ f) = \mathrm{cod}(g)$, i.e. $g \circ f\colon \mathrm{dom}(f) \to \mathrm{cod}(g)$.
  5. For every object A an arrow $1_A\colon A \to A$ called the identity morphism, such that for any $f\colon A \to B$ we have $f \circ 1_A = f = 1_B \circ f$.

  We write the class of arrows $f\colon A \to B$ as $\mathrm{Mor}_{\mathcal{C}}(A, B)$. The notations $\mathrm{Hom}_{\mathcal{C}}(A, B)$, $\mathcal{C}(A, B)$, $[A, B]_{\mathcal{C}}$, and $(A, B)_{\mathcal{C}}$ are also used.
- **Observation 4.18.** *Many classes of mathematical objects and their natural (structure-preserving) mappings form categories.*
- **Definition 4.19.** Category theory studies general properties of structures abstracting away from the concrete objects.

Categories in KRMT

- **Remark:** We have already seen various examples of categories in KRMT
- **Example 4.20.** Types and functions in MMT/LF form a category. (abstract away from terms)
- **Example 4.21.** Contexts and substitutions in logics form a category: A substitution σ induces a function from $\mathit{wff}(\Sigma, \Gamma \uplus \mathrm{supp}(\sigma))$ to $\mathit{wff}(\Sigma, \Gamma \uplus \mathrm{intro}(\sigma))$.
- **Example 4.22.** MMT theories and theory morphisms form a category: A theory T defines a language (set of well typed terms) $\mathcal{L}_T$, and a theory morphism from S to T mapping between $\mathcal{L}_S$ and $\mathcal{L}_T$.

Commonly used Categories

- **Definition 4.23.** The objects of the category of sets **Set** are sets and its arrows $f\colon A \to B$ are the functions.
- **Definition 4.24.** The objects of the category of topological spaces **Top** are topological spaces and its arrows are the continuous functions.
- **Definition 4.25.** A category $\mathcal{C}$ is called small (otherwise large), iff $\mathrm{ob}(\mathcal{C})$ and $\mathrm{Mor}_{\mathcal{C}}$ consist of sets (not classes).
- **Definition 4.26.** Let $\mathcal{C}$ be a category, then the opposite category (also called the dual category) $\mathcal{C}^{\mathrm{op}}$ is formed by reversing all the arrows of $\mathcal{C}$, i.e.

$$\mathrm{Mor}_{\mathcal{C}^{\mathrm{op}}} := \{f\colon B \to A \mid f\colon A \to B \in \mathrm{Mor}_{\mathcal{C}}\}$$

Functors

- **Definition 4.27.** Let $\mathcal{C}$ and $\mathcal{D}$ be categories, then a mapping F from $\mathcal{C}$ to $\mathcal{D}$ is called a (covariant) functor, iff F
  - associates to each $X \in \mathrm{ob}(\mathcal{C})$ an object $F(X) \in \mathrm{ob}(\mathcal{D})$
  - associates to each morphism $f\colon X \to Y \in \mathrm{Mor}_{\mathcal{C}}(X, Y)$ a morphism

  $$F(f)\colon F(X) \to F(Y) \in \mathrm{Mor}_{\mathcal{D}}(F(X), F(Y))$$

  such that the following two conditions hold:
  - $F(1_X) = 1_{F(X)}$ for each $X \in \mathrm{ob}(\mathcal{C})$.
  - $F(g \circ f) = F(g) \circ F(f)$ for all morphisms $f\colon X \to Y$ and $g\colon Y \to Z$ in $\mathcal{C}$.

  That is, functors must preserve identity morphisms and morphism composition.
- **Definition 4.28.** The category of small categories (denoted as **Cat**) has all small categories as objects and functors as arrows.
- **Observation 4.29.** ***Cat*** *is itself a large category.*

1.4.2 Example/Motivation: Natural Numbers in Category Theory

Lawvere's Natural Numbers Object

- **Recap:** In set theory, we define the natural numbers by the five Peano axioms about $\mathbb{N}$, $0\in\mathbb{N}$, and $s\colon\mathbb{N}\to\mathbb{N}$.
- In category theory we can give a different answer! (need more terminology)
- **Definition 4.30.** A natural number object (NNO) in a (Cartesian closed) category E with terminal object 1 is an object $\mathbb{N}$ in E equipped with
  - a morphism $z\colon 1\to\mathbb{N}$ from the terminal object 1 (zero)
  - a morphism $s\colon\mathbb{N}\to\mathbb{N}$ (successor)

  such that for every other diagram $1\xrightarrow{q}A\xrightarrow{f}A$ there is a unique morphism $u\colon\mathbb{N}\to A$ such that the following diagram commutes:

$$\begin{tikzcd}
1 \arrow[r, "z"] \arrow[dr, "q"'] & \mathbb{N} \arrow[r, "s"] \arrow[d, dashed, "u"] & \mathbb{N} \arrow[d, dashed, "u"] \\
 & A \arrow[r, "f"] & A
\end{tikzcd}$$

Natural Numbers $\widehat{=}$ natural number object in **Set**

- **Theorem 4.31.** *The natural number object in* ***Set*** *is isomorphic to Peano's* $\mathbb{N}$*.*
- Peano's $\mathbb{N}$ by the Recursion Theorem [ML86, §II.3].
- **Lemma 4.32.** *The natural number object* $\langle\mathbb{N}, z, s\rangle$ *in* ***Set*** *obeys Peano's axioms.*
- *Proof:*
  1. For **P1** note that 1 in **Set** is a singleton set $\{a\}$, and any function $z\colon 1\to\mathbb{N}$ identifies an element $z(a)$ (let's call it z as well) in $\mathbb{N}$.
  2. For **P2** note that s in **Set** is a function.
  3. For **P3** assume $s(n) = z$ and consider a diagram $1 \xrightarrow{e} A \xrightarrow{f} A$ with $A = \{e, d\}$ and $u(e) = u(d) = d$. Then there is a function $f\colon \mathbb{N}\to A$ such that $f(z) = e$ and $f(s(n)) = u(f(n))$. But if $s(n) = z$ then $f(s(n)) = e \neq d = u(f(n))$.
  4. Injectivity of s (**P4**) is left as an exercise.
  5. **P5**, see **??**

The Language of Diagrams

- **Definition 4.33.** A diagram in a category E is a directed graph, where the nodes are objects of E and the edges are arrows of E connecting the respective objects. Diagrams often use dashed arrows to signify unique existence of arrows.
- **Definition 4.34.**
  Let D be a diagram, then we say that D commutes (or is commuative), iff for any two paths $f_1, \ldots, f_n$ and $g_1, \ldots, g_m$ with the same start and end in D we have $f_n \circ \ldots \circ f_1 = g_m \circ \ldots \circ g_1$.
- **Example 4.35.**

  Let $f\colon A \to B$, $g\colon A \to C$, $u\colon C \to D$, and $v\colon B \to D$ in a category $\mathcal{C}$, then we say that the diagram on the right commutes, iff $f \circ v = g \circ u$.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"] & B \arrow[d, "v"] \\
C \arrow[r, "u"] & D
\end{tikzcd}$$

- **Definition 4.36.**

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "g"] & B \arrow[d, "u"] \\
 & D
\end{tikzcd}$$

We treat the left diagram as an abbreviation of the right one.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "1_A"] & B \arrow[d, "u"] \\
A \arrow[r, "g"] & D
\end{tikzcd}$$

Diagram Chase: the Proof Method in Category Theory

- **Definition 4.37 (Diagram Chase in Small Categories with Functions).**

  If $\mathcal{C}$ is small and f, g, u, and v are functions (e.g. in In **Set**), the diagram above commutes, iff the commutativity equation $v(f(a)) = u(g(a))$ holds for all $a \in A$.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"] & B \arrow[d, "v"] \\
C \arrow[r, "u"] & D
\end{tikzcd}$$

  We use the commutativity equation (and other properties of arrows) in the proof method of diagram chase (or diagrammatic search), which involves “chasing” elements around the diagram, until the desired element or result is constructed or verified.

- **Example 4.38.**

  The diagram on the right commutes, iff $k(g(f(x))) = k(h(x)) = g'(f'(f(x)))$ for all $x \in X$.

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, "h"'] & Y \arrow[r, "f'"] \arrow[d, "g"] & Y' \arrow[d, "g'"] \\
 & Z \arrow[r, "k"] & Z'
\end{tikzcd}$$

Natural Number Objects in **Set**: Induction I

- **Lemma 4.39.** *The natural number object in* ***Set*** *is inductive: If $A \subseteq \mathbb{N}$ and from $z \in \mathbb{N}$ and $a \in A$ we obtain $s(a) \in A$ we obtain $A = \mathbb{N}$.*
- *Proof:* We translate the assumptions to diagrams and conduct a diagram chase.
  1. We extend the NNO diagram with an inclusion function $i \colon A \to \mathbb{N}$ that corresponds to $A \subseteq \mathbb{N}$. Note that every cell commutes in the diagram on the left.

$$\begin{tikzcd}
1 \arrow[r, "z"] \arrow[d, "1_1"'] & \mathbb{N} \arrow[r, "s"] \arrow[d, dashed, "u"] & \mathbb{N} \arrow[d, dashed, "u"] \\
1 \arrow[r, "z"] \arrow[d, "1_1"'] & A \arrow[r, "s|_A"] \arrow[d, "i"] & A \arrow[d, "i"] \\
1 \arrow[r, "z"] & \mathbb{N} \arrow[r, "s"'] & \mathbb{N}
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, "z"] \arrow[d, "1_1"'] & \mathbb{N} \arrow[r, "s"] \arrow[d, "1_{\mathbb{N}}"] & \mathbb{N} \arrow[d, "1_{\mathbb{N}}"] \\
1 \arrow[r, "z"] & \mathbb{N} \arrow[r, "s"'] & \mathbb{N}
\end{tikzcd}$$

     Note that $s|_A \colon A \to A$ as $a \in A$ implies $s(a) \in A$. (induction step assumption)
  2. Trivially, also the diagram on the right commutes, so by uniqueness in NNO, we have $i \circ u = 1_{\mathbb{N}}$.
  3. Given two composable functions f and g, if $f \circ g$ is the identity, then f is injective.

Natural Number Objects in **Set**: Induction II

4. So $U\colon \mathbb{N}\to A$ is injective, in other words: $\mathbb{N} \subseteq A$, and thus $A = \mathbb{N}$.

Uniqueness of Natural Numbers

- **Theorem 4.40.** *The natural number object is uniquely determined up to isomorphism in a category.*
- *Proof:* We prove that if there is another diagram $1 \xrightarrow{z'} \mathbb{N}' \xrightarrow{s'} \mathbb{N}'$, then $\mathbb{N}$ and $\mathbb{N}'$ are isomorphic.
  1. We show that there are functions $f \colon \mathbb{N} \to \mathbb{N}'$ and $f' \colon \mathbb{N}' \to \mathbb{N}$, such that $f \circ f' = \mathrm{Id}_{\mathbb{N}'}$ and $f' \circ f = \mathrm{Id}_{\mathbb{N}}$.
  2. We have the following two commuting diagrams

$$\begin{tikzcd}
1 \arrow[r, "z"] \arrow[d, "1_1"'] & \mathbb{N} \arrow[r, "s"] \arrow[d, dashed, "f"] & \mathbb{N} \arrow[d, dashed, "f"] \\
1 \arrow[r, "z'"] \arrow[d, "1_1"'] & \mathbb{N}' \arrow[r, "s'"] \arrow[d, dashed, "f'"] & \mathbb{N}' \arrow[d, dashed, "f'"] \\
1 \arrow[r, "z"] & \mathbb{N} \arrow[r, "s"'] & \mathbb{N}
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, "z"] \arrow[d, "1_1"'] & \mathbb{N} \arrow[r, "s"] \arrow[d, "1_{\mathbb{N}}"] & \mathbb{N} \arrow[d, "1_{\mathbb{N}}"] \\
1 \arrow[r, "z"] & \mathbb{N} \arrow[r, "s"'] & \mathbb{N}
\end{tikzcd}$$

     The left one comes from the universal property of $1 \xrightarrow{z} \mathbb{N} \xrightarrow{s} \mathbb{N}$ and $1 \xrightarrow{z'} \mathbb{N}' \xrightarrow{s'} \mathbb{N}'$, the right one by construction. hence $f' \circ f = 1_{\mathbb{N}}$.
  3. We obtain $f \circ f' = 1_{\mathbb{N}'}$ by a similar argument.

1.4.3 Universal Constructions in Category Theory

Initial and Terminal Objects

- **Definition 4.41.** Let $\mathcal{C}$ be a category, then we call an object $I{\in}\mathrm{ob}(\mathcal{C})$ initial (also cofinal or universal and written as 0), iff for every $X{\in}\mathrm{ob}(\mathcal{C})$ there is exactly one arrow $a\colon I{\to}X$. If every arrow into I is an isomorphism, then I is called strict initial object.

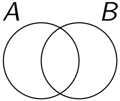
  Definition 4.42. An object $T{\in}\mathrm{ob}(\mathcal{C})$ is called terminal or final, iff for every $X{\in}\mathrm{ob}(\mathcal{C})$ there is exactly one arrow $a\colon X{\to}T$. A terminal object is also called a terminator and write it as 1.
- **Observation 4.43.** *Initial and terminal objects are unique up to isomorphism, if they exist at all.* *(they need not exist in all categories)*
- **Example 4.44.** In **Set** the initial object is the empty set, while the terminal object is the (unique up to isomorphism) singleton set.
- **Remark:** We can think of the initial and terminal objects the category-theoretic generalizations (“universal characterizations”) of the empty and singleton sets: they are characterized by objects and arrows only.

Pushouts: Unions on Steroids

- **Question:** Can we also characterize operations like union universally?

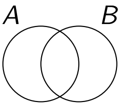
Pushouts: Unions on Steroids

- **Question:** Can we also characterize operations like union universally?
- **Idea:** In $A \cup B$, we use $A \cap B$ twice.
  We have $A \cap B \subseteq A$ and $A \cap B \subseteq B$, which we can express with arrows (inclusions) $A \cap B \xhookrightarrow{\iota_A} A$ and $A \cap B \xhookrightarrow{\iota_B} B$. Similarly we have $A \subseteq A \cup B$ and $B \subseteq A \cup B$ which we express as $A \xhookrightarrow{\iota_A} A \cup B$ and $B \xhookrightarrow{\iota_B} A \cup B$.



Pushouts: Unions on Steroids

- **Question:** Can we also characterize operations like union universally?
- **Idea:** In $A \cup B$, we use $A \cap B$ twice.
  We have $A \cap B \subseteq A$ and $A \cap B \subseteq B$, which we can express with arrows (inclusions) $A \cap B \xhookrightarrow{\iota_A} A$ and $A \cap B \xhookrightarrow{\iota_B} B$. Similarly we have $A \subseteq A \cup B$ and $B \subseteq A \cup B$ which we express as $A \xhookrightarrow{\iota_A} A \cup B$ and $B \xhookrightarrow{\iota_B} A \cup B$.



- **Definition 4.47.**Let $\mathcal{C}$ be a category, then the pushout of morphisms $f\colon Z \to X$ and $g\colon Z \to Y$ consists of an object P together with two morphisms $i_f\colon X \to P$ and $i_g\colon Y \to P$, such that the left diagram below commutes and that $\langle P, i_f, i_g\rangle$ is universal with respect to this diagram – i.e., for any other such set $\langle Q, i_f, i_g\rangle$ for which the following diagram commutes, there must exist a unique $u\colon P \to Q$ also making the diagram commute, i.e.

$$\begin{tikzcd}
Y \arrow[r, "i_g"] & P \\
Z \arrow[u, "g"'] \arrow[r, "f"] & X \arrow[u, "i_f"']
\end{tikzcd}$$

$$\begin{tikzcd}
 & & Q \\
Y \arrow[r, "i_g"'] \arrow[urr, bend left, "i_g"] & P \arrow[ur, dashed, "u"] & \\
Z \arrow[u, "g"'] \arrow[r, "f"] & X \arrow[u, "i_f"] \arrow[uur, bend right, "i_f"'] &
\end{tikzcd}$$

Pushouts in **Set**

- As with all universal constructions, the pushout, if it exists, is unique up to a unique isomorphism.
- If X, Y, and Z are sets, and $f\colon Z \to X$ and $g\colon Z \to Y$ are function, then the pushout of f and g is the disjoint union $X \uplus Y$, where elements sharing a common preimage (in Z) are identified, i.e. $P = (X \uplus Y)/\sim$, where $\sim$ is the finest equivalence relation such that $\iota_1(f(z)) \sim \iota_2(g(z))$.
- **In particular:** if $X, Y \subseteq W$ for some larger set W, $Z = X \cap Y$, and f and g the inclusions of Z into X and Y, then the pushout can be canonically identified with $X \cup Y$.

Product Objects and Exponentials in Categories

- **Question:** Can we also characterize functions (function spaces) in categories?

Product Objects and Exponentials in Categories

- **Question:** Can we also characterize functions (function spaces) in categories?
- **Idea:** Functions are sets of pairs with additional properties (left totality and right uniqueness)

Product Objects and Exponentials in Categories

- **Question:** Can we also characterize functions (function spaces) in categories?
- **Idea:** Functions are sets of pairs with additional properties (left totality and right uniqueness)
- **Definition 4.50.** Let $\mathcal{C}$ be a category and $X_1, X_2 \in \mathrm{ob}(\mathcal{C})$. Then we call an object X together with two morphisms $\pi_1 \colon X \to X_1$ and $\pi_2 \colon X \to X_2$ the product of X_1 and X_2 and write it as $X_1 \times X_2$ if it satisfies the following universal property: For every object Y and pair of morphisms $f_1 \colon Y \to X_1$ and $f_2 \colon Y \to X_2$ there exists a unique morphism $f \colon Y \to X_1 \times X_2$ such that the diagram on the right commutes:

$$\begin{tikzcd} & Y \arrow[dl, "f_1"'] \arrow[d, dashed, "f"'] \arrow[dr, "f_2"] & \\ X_1 & X_1 \times X_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & X_2 \end{tikzcd}$$

The unique morphism f is called the product of morphisms f_1 and f_2 and is denoted $\langle f_1, f_2 \rangle$. The morphisms π_1 and π_2 are called the (canonical) projection or projection morphism.

Products in Set and Top

- **Example 4.51.** In **Set**, the product is the Cartesian product: Given sets X_1 and X_2, then we have the projections $\pi_i\colon X_1 \times X_2 \to X_i$. Given any set Y with functions $f_i\colon Z \to X_i$, the universal arrow f is defined as $f\colon Y \to X_1 \times X_2; y \mapsto \langle f_1(y), f_1(y)\rangle$.
- **Example 4.52.**
  In **Top**, the product of two objects is the product topology.

Exponentials in Categories

- **Definition 4.53.** If $A \times B$ exists for all objects A and B in a category $\mathcal{C}$, then we say that $\mathcal{C}$ has all binary products.
- **Definition 4.54.** Let $\mathcal{C}$ be a category that has all binary products and $Z, Y \in \mathrm{ob}(\mathcal{C})$, then we call an object Z^Y together with a morphism $\mathrm{eval}\colon Z^Y \times Y \to Z$ is called an exponential object, iff for any $X \in \mathrm{ob}(\mathcal{C})$ and $g\colon X \times Y \to Z \in \mathrm{Mor}_{\mathcal{C}}$ there is a unique morphism $\lambda g\colon X \to Z^Y$ (called the transpose of g) such that the following diagram commutes:

$$\begin{tikzcd}
X \arrow[d, dashed, "\lambda g"'] & X \times Y \arrow[d, dashed, "\langle \lambda g, 1_Y \rangle"'] \arrow[dr, "g"] & \\
Z^Y & Z^Y \times Y \arrow[r, "\mathrm{eval}"'] & Z
\end{tikzcd}$$

- **Lemma 4.55.** *In* ***Set***, $Z^Y = Y \to Z$ *and* $\mathit{eval}\colon Z^Y \times Y \to Z; (f,y) \mapsto f(y)$. *For any map* $g\colon X \times Y \to Z$ *the map* $\lambda g\colon X \to Z^Y$ *is the Curried form of* g*:* $\lambda g(x)(y) = g(x,y)$.

Cartesian Closed Categories

- **Definition 4.56.** A category $\mathcal{C}$ is called Cartesian closed (a CCC), iff it satisfies the following three properties:
  - $\mathcal{C}$ has a terminal object.
  - Any two objects X and Y of $\mathcal{C}$ have a product $X \times Y$ in $\mathcal{C}$.
  - Any two objects Y and Z of $\mathcal{C}$ have an exponential Z^Y in $\mathcal{C}$.

1.5 Axiomatic Set Theory (ZFC)

1.5.1 Naive Set Theory

(Naive) Set Theory [Can95; Can97]

- **Definition 5.1.** A set is "everything that can form a unity in the face of God". (Georg Cantor ($*$1845, †1918))
- **Example 5.2.** (determination by elementhood relation $\in$)
  - "the set that consists of the number 7 and the prime divisors of 510510"
  - $\{7, c\}$, $\{1, 2, 3, 4, 5n, \ldots\}$, $\{x | x \text{ is an integer}\}$, $\{X | \mathrm{P}(X)\}$
- **Questions (extensional/intensional):**
  - If $c = 7$, is $\{7, c\} = \{7\}$?
  - Is $\{X | X \in \mathbb{N}, X \neq X\} = \{X | X \in \mathbb{N}, X^2 < 0\}$?
  - yes $\rightsquigarrow$ *extensional*; no $\rightsquigarrow$ *intensional*;

(Naive) Set Theory: Formalization

- **Idea:** Use first-order logic (with equality)
  - Signature: $\Sigma{:=}\{\in\ldots\}$ (sets are individuals)
  - Extensionality: $\forall M,N.M=N\Leftrightarrow(\forall X.(X{\in}M)\Leftrightarrow(X{\in}N))$(two sets are equal, iff they have the same elements)
  - Comprehension: $\exists M.\forall X.(X{\in}M)\Leftrightarrow \mathrm{E}$ (all sets that we can write down exist)
  - **Note**: The comprehension axiom is schematic in expression E!
- **Idea:** Define set theoretic concepts from $\in$ as signature extensions

Union	$\cup{\in}\Sigma_2^f$	$\forall M,N,X.(X{\in}(M\cup N))\Leftrightarrow(X{\in}M\vee X{\in}N)$
Intersection	$\cap{\in}\Sigma_2^f$	$\forall M,N,X.(X{\in}(M\cap N))\Leftrightarrow(X{\in}M\wedge X{\in}N)$
Empty set	$\emptyset{\in}\Sigma_0^f$	$\neg(\exists X.X{\in}\emptyset)$
and so on.	$\vdots$	$\vdots$

(Naive) Set Theory (Problems)

- **Example 5.3 (The set of all set and friends).** $\{M|M \text{ set}\}$, $\{M|M \text{ set}, M \in M\}$, ...
- **Definition 5.4 (Problem).** Russell's Antinomy:

$$\mathcal{M} := \{M|M \text{ set}, M \notin M\}$$

  the set $\mathcal{M}$ of all sets that do not contain themselves.
- **Question:** Is $\mathcal{M} \in \mathcal{M}$? **Answer**: $\mathcal{M} \in \mathcal{M}$ iff $\mathcal{M} \notin \mathcal{M}$.
- **What happened?:** We have written something down that makes problems
- **Solutions: Define away the problems:**

weaker comprehension	axiomatic set theory	now
weaker properties	higher-order logic	done
non-standard semantics	domain theory [Scott]	another time

1.5.2 ZFC Axioms

Axiomatic Set Theory in First-Order Logic

- **Idea:** Avoid paradoxes by cautious (*axiomatic*) comprehension. ([Zer08])

Ex	$\exists X.X = X$	There is a set
Ext	$\forall M, N.M = N \Leftrightarrow (\forall X.(X \in M) \Leftrightarrow (X \in N))$	Extensionality
Sep	$\forall N.\exists M.\forall Z.(Z \in M) \Leftrightarrow (Z \in N \wedge \mathrm{E})$ From a given set N we can separate all members described by expression E. (which may contain Z)	

- **Theorem 5.5.** $\forall M, N.(M \subseteq N) \wedge (N \subseteq M) \Rightarrow M = N$
- **Theorem 5.6.** *M is uniquely determined in* Sep

  Proof sketch: With Ext

- **Notation:** Write $\{X \in N \mid \mathrm{E}\}$ for the set M guaranteed by Sep.

Quality Control

- **Question:** Is *ZFC* good? (make this more precise under various views)
  foundational: Is ZFC sufficient for mathematics?
  adequate: is the ZFC notion of sets adequate?
  formal: is ZFC consistent?
  ambitious: Is ZFC complete?
  pragmatic: Is the formalization convenient?
  computational: does the formalization yield computation-guiding structure?
- Questions like these help us determine the quality of a foundational system or theory.

How about Russel's Antinomy?

- **Theorem 5.7.** *There is no universal set.*
- *Proof:*
  1. For each set M, there is a set $M_R := \{X \in M \mid X \notin X\}$ by Sep.
  2. Show $\forall M. M_R \notin M$.
  3. If $M_R \in M$, then $M_R \notin M_R$, (also if $M_R \notin M$)
  4. Thus $M_R \notin M$ or $M_R \in M_R$.
- **Intuition:** To get the paradox we would have to separate from the universal set $\mathcal{A}$, to get $\mathcal{A}_R$.
- **Great,** then we can continue developing our set theory!

Are there Interesting Sets at all?

- **Question:** Are there Interesting Sets at all?
- **Answer:** Yes, e.g. the empty set:
  - Let M be a set (there is one by Ex; we do not need to know what it is)
  - Define $\emptyset{:=}\{X{\in}M\,|\,X \neq X\}$.
  - $\emptyset$ is empty and uniquely determined by Ext.
- **Even more:** Intersections: $M \cap N{:=}\{X{\in}M\,|\,X{\in}N\}$
- **Question:** How about $M \cup N$? or $\mathbb{N}$?
- **Answer:** we do not know they exist yet! (need more axioms)
  Hint: consider $\mathcal{D}_\iota = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \ldots\}$

Is Set theory enough? ⇝ Nicolas Bourbaki

- Is it possible to develop all of Mathematics from set theory?
  ⇝ N. Bourbaki: Éléments de Mathématique~~s~~ (there is only one mathematics)
- **Original Goal:** A modern textbook on calculus.
- **Result:** 40 volumes in nine books from 1939 to 1968

  Set Theory [Bou68]
  Algebra [Bou74]
  Topology [Bou89]
  Functions of one real variable
  Integration
  Topological Vector Spaces
  Commutative Algebra
  Lie Theory
  Spectral Theory
- **Contents:**
  - Starting from set theory all of the fields above are developed.
  - All proofs are carried out, no references to other books.

The Axioms for Set Union

- **Axiom 5.8 (Small Union Axiom $\cup$Ax).** *For any sets M and N there is a set W, that contains all elements of M and N.* $\forall M, N.\exists W.\forall X.(X{\in}M \vee X{\in}N) \Rightarrow X{\in}W$
- **Definition 5.9.** $M \cup N{:=}\{X{\in}W \mid X{\in}M \vee X{\in}N\}$ (exists by Sep.)
- **Axiom 5.10 (Large Union Axiom $\bigcup$Ax).** *For each set M there is a set W, that contains the elements of all elements of M.* $\forall M.\exists W.\forall X, Y.Y{\in}M \Rightarrow X{\in}Y \Rightarrow X{\in}W$
- **Definition 5.11.** $(\bigcup M){:=}\{X | \exists Y.Y{\in}M \wedge X{\in}Y\}$ (exists by Sep.)
- This also gives us intersections over families (without another axiom):
- **Definition 5.12.**

$$(\bigcap M){:=}\{Z{\in}\bigcup M \mid \forall X.X{\in}M \Rightarrow Z{\in}X\}$$

The Power Set Axiom

- **Axiom 5.13 (Power Set Axiom).** *For each set M there is a set W that contains all subsets of M:* $\wp Ax{:=}(\forall M.\exists W.\forall X.(X\subseteq M)\Rightarrow X\in W)$
- **Definition 5.14.** Power Set: $\wp(M){:=}\{X|X\subseteq M\}$ (Exists by Sep.)
- **Definition 5.15.** Singleton set: $\{X\}{:=}\{Y\in\wp(X)\,|\,X=Y\}$
- **Axiom 5.16 (Pair Set (Axiom)).** *(is often assumed instead of* $\cup$Ax*)*
  Given sets M and N there is a set W that contains exactly the elements M and N: $\forall M,N.\exists W.\forall X.(X\in W)\Leftrightarrow((X=N)\vee(X=M))$
- Is derivable from $\wp$Ax: $\{M,N\}{:=}\{M\}\cup\{N\}$.
- **Definition 5.17 (Finite Sets).** $\{X,Y,Z\}{:=}\{X,Y\}\cup\{Z\}\ldots$
- **Theorem 5.18.** $\forall Z,X_1,\ldots,X_n.(Z\in\{X_1,\ldots,X_n\})\Leftrightarrow(Z=X_1\vee\ldots\vee Z=X_n)$

The Foundation Axiom

- **Axiom 5.19 (The Foundation Axiom Fund).** *Every non-empty set has a ∈-minimal element,.* $\forall X.(X \neq \emptyset) \Rightarrow (\exists Y.Y{\in}X \wedge \neg(\exists Z.Z{\in}X \wedge Z{\in}Y))$
- **Theorem 5.20.** *There are no infinite descendig chains* $\ldots, X_2, X_1, X_0$ *and thus no cycles* $\ldots X_1, X_0, \ldots, X_2, X_1, X_0$.
- **Definition 5.21.** Fund guarantees a hierarchical structure (von Neumann Hierarchy) of the universe.
  1. 0. order: $\emptyset$,
  2. 1. order: $\{\emptyset\}$,
  3. 2. order: all subsets of 1. order, $\cdots$
- **Note:** In contrast to a Russel-style typing where sets of differernt type are distinct, this categorization is cummulative.

The Infinity Axiom

- We already know a lot of sets
  - e.g. $\emptyset$, $\{\emptyset\}$, $\{\{\emptyset\}\}$, … (iterated singleton set)
  - or $\emptyset$, $\{\emptyset\}$, $\{\emptyset, \{\emptyset\}\}$, … (iterated pair set)

  But Does the set $\mathbb{N}$ of all members of these sequences?
- **Axiom 5.22 (Infinity Axiom ∞Ax).** *There is a set that contains $\emptyset$ and with each X also $X \cup \{X\}$.* $\exists M.\emptyset \in M \wedge (\forall Z.Z \in M \Rightarrow (Z \cup \{Z\}) \in M)$.
- **Definition 5.23.** M is inductive: $\mathrm{Ind}(M) := \emptyset \in M \wedge (\forall Z.Z \in M \Rightarrow (Z \cup \{Z\}) \in M)$.
- **Definition 5.24.** Set of the Inductive Set: $\omega := \{Z | \forall W.\mathrm{Ind}(W) \Rightarrow Z \in W\}$
- **Theorem 5.25.** *ω is inductive.*

The Replacement Axiom

- We have ω, $\wp(M)$, but not $\{\omega, \wp(\omega), \wp(\wp(\omega)), \ldots\}$.
- **Axiom 5.26 (The Replacement Axiom (Schema): Rep).**
  If for each X there is exactly one Y with property $\mathrm{P}(X, Y)$*, then for each set U, that contains these X, there is a set V that contains the respective Y.*
  $(\forall X.\exists^1 Y.\mathrm{P}(X, Y)) \Rightarrow (\forall U.\exists V.\forall X, Y.X{\in}U \wedge \mathrm{P}(X, Y) \Rightarrow Y{\in}V)$
- **Intuitively:** A right-unique property P induces a replacement $\forall U.\exists V.V = \{F(X)|X{\in}U\}$.
- **Example 5.27.** Let $U = \{1, \{2, 3\}\}$ and $\mathcal{P}(X \Leftrightarrow Y) \Leftrightarrow (\forall Z.Z{\in}Y \Rightarrow Z = X)$, then the induced function F maps each X to the set V that contains X, i.e. $V = \{\{X\}|X{\in}U = \{\{1\}, \{\{2, 3\}\}\}\}$.

Zermelo Fraenkel Set Theory

- **Definition 5.28 (Zermelo Fraenkel Set Theory).**
  We call the first-order theory given by the axioms below Zermelo/Fraenkel set theory and denote it by ZF.

Ex	$\exists X.X = X$
Ext	$\forall M, N.M = N \Leftrightarrow (\forall X.(X\in M) \Leftrightarrow (X\in N))$
Sep	$\forall N.\exists M.\forall Z.(Z\in M) \Leftrightarrow (Z\in N \wedge \mathrm{E})$
$\cup$Ax	$\forall M, N.\exists W.\forall X.(X\in M \vee X\in N) \Rightarrow X\in W$
$\bigcup$Ax	$\forall M.\exists W.\forall X, Y.Y\in M \Rightarrow X\in Y \Rightarrow X\in W$
$\wp$Ax	$\forall M.\exists W.\forall X.(X\subseteq M) \Rightarrow X\in W$
∞Ax	$\exists M.\emptyset\in M \wedge (\forall Z.Z\in M \Rightarrow (Z \cup \{Z\})\in M)$
Rep	$(\forall X.\exists^1 Y.\mathrm{P}(X, Y)) \Rightarrow (\forall U.\exists V.\forall X, Y.X\in U \wedge \mathrm{P}(X, Y) \Rightarrow Y\in V)$
Fund	$\forall X.(X \neq \emptyset) \Rightarrow (\exists Y.Y\in X \wedge \neg(\exists Z.Z\in X \wedge Z\in Y))$

The Axiom of Choice

- **Axiom 5.29 (The axiom of Choice :AC).**
  For each set X of non-empty, pairwise disjoint subsets there is a set that contains exactly one element of each element of X.
  $\forall X, Y, Z. Y \in X \wedge Z \in X \Rightarrow ((Y \neq \emptyset) \wedge (Y = Z \vee Y \cap Z = \emptyset) \Rightarrow (\exists . \forall . V \in X \Rightarrow (\exists . U \cap V = \{\})))$
- This axiom assumes the existence of a set of representatives, even if we cannot give a construction for it. $\rightsquigarrow$ we can “pick out” an arbitrary element.
- **Reasons for AC:**
  - Neither $\mathrm{ZF} \vdash \mathrm{AC}$, nor $\mathrm{ZF} \vdash \neg\mathrm{AC}$
  - So it does not harm?
- **Definition 5.30 (Zermelo Fraenkel Set Theory with Choice).**
  The theory ZF together with AC is called ZF with choice and denoted as ZFC.

1.5.3 ZFC Applications

Limits of ZFC

- There is no set whose cardinality is strictly between that of integers and real numbers.
- **Theorem 5.31.**
  If ZFC is consistent, then neither CH nor ¬CH can be derived. (*CH is independent of ZFC*)
- The axiomatzation of ZFC does not suffice.
- There are other examples like this.

Ordered Pairs

- **Empirically:** In ZFC we can define all mathematical concepts.
- **For Instance:** We would like a set that behaves like an odererd pair.
- **Definition 5.32.** Define $\langle X,Y\rangle{:=}\{\{X\},\{X,Y\}\}$
- **Lemma 5.33.** $\langle X,Y\rangle = \langle U,V\rangle \Rightarrow X = U \wedge Y = V$
- **Lemma 5.34.** $U{\in}X \wedge V{\in}Y \Rightarrow \langle U,V\rangle{\in}\wp(\wp(X\cup Y))$
- **Definition 5.35.** Left projection: $\pi_l(X) = \begin{cases} U & \text{if } (\exists V.X = \langle U,V\rangle) \\ \emptyset & \text{if } X \text{ is no pair} \end{cases}$
- **Definition 5.36.** Right projection π_r analogous.

Relations

- All mathematical objects are represented by sets in ZFC, in particular relations
- **Definition 5.37.** The Cartesian product of X and Y
  $X \times Y := \{Z \in \wp(\wp(X \cup Y)) \mid Z \text{ is ordered pair with } \pi_l(Z) \in X \wedge \pi_r(Z) \in Y\}$
  A relation is a subset of a Cartesian product.
- **Definition 5.38.** The domain and codomain of a function are defined as usual:

$$\mathrm{Dom}(X) := \begin{cases} \{\pi_l(Z) | Z \in X\} & \text{if } X \text{ is a relation} \\ \emptyset & \text{else} \end{cases}$$

$$\mathrm{coDom}(X) := \begin{cases} \{\pi_r(Z) | Z \in X\} & \text{if } X \text{ is a relation} \\ \emptyset & \text{else} \end{cases}$$

  but they (as first-order functions) must be total, so we (arbitrarily) extend them by the empty set for non-relations

Functions

- **Definition 5.39.** A function f from X to Y is a right unique relation with $\mathrm{Dom}(f) = X$ and $\mathrm{coDom}(f) = Y$; write $f \colon X \to Y$.
- **Definition 5.40.** function application:

$$f(X) = \begin{cases} Y & \text{if } f \text{ function and } (\langle X, Y \rangle \in f) \\ \emptyset & \text{else} \end{cases}$$

Domain Language vs. Representation Language

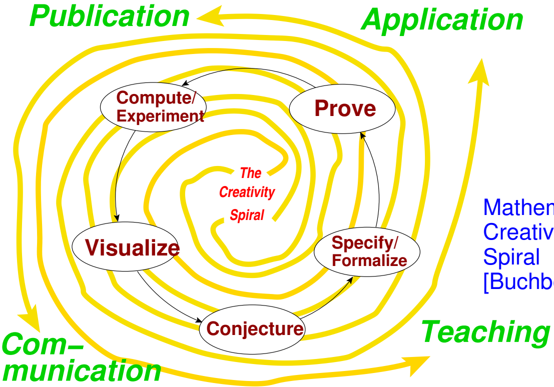
- **Note:** Relations and functions are objects of set theory, $ZFC\in$ is a predicate of the representation language.
- Predicates and functions of the representation language can be expressed in the object language:
  - $\forall A.\exists R.R = \{\langle U, V\rangle | U\in A \wedge V\in A \wedge p(U \wedge V)\}$ for all predicates p.
  - $\forall A.\exists F.F = \{\langle X, f(X)\rangle | X\in A\}$ for all functions f.
- As the natural numbers can be epxressed in set theory, the logical calculus can be expressed by Gödelization.

Chapter 2
Aspects of Knowledge Representation for Mathematics

2.1 Project Tetrapod

The way we do math will change dramatically

- **Definition 1.1 (Doing Math).** Buchberger's Math creativity spiral



Mathematical Creativity Spiral [Buchberger 1995]

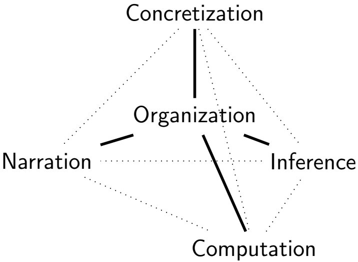
- Every step will be supported by mathematical software systems
- Towards an infrastructure for web-based mathematics!

Knowledge Representation is only Part of "Doing Math"

- **Definition 1.2.** One of the key insights is that the mathematics ecosystem involves a body of knowledge externalized in an ontology that provides organization and combines the following four aspects:
  - Inference: exploring theories, formulating conjectures, and constructing proofs
  - Computation: simplifying mathematical objects, re contextualizing conjectures. . .
  - Concretization: collecting concrete examples/models, applying mathematical knowledge to real-world problems and situations.
  - Narration: devising both informal and formal languages for expressing mathematical ideas, visualizing mathematical data, presenting mathematical developments, organizing and interconnecting mathematical knowledge

“Doing Math”: as a Tetrapod

- We call the endeavour of creating a computer-supported mathematical ecosystem “Project tetrapod” as it needs to stand on four legs.



- **Collaborators:** KWARC@FAU, McMaster University

2.2 The Flexiformalist Program: Introduction

Background: Mathematical Documents

- Mathematics plays a fundamental role in Science, Technology, and Engineering (learn from Math, apply for STEM)
- Mathematical knowledge is rich in content, sophisticated in structure, and technical in presentation,
- its conservation, dissemination, and utilization constitutes a challenge for the community and an attractive line of inquiry.
- **Challenge:** How can/should we do mathematics in the 21st century?
- Mathematical knowledge and objects are transported by documents
- **Three levels of electronic documents:**

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

  transforming down is simple, transforming up needs humans or AI.
- **Observation:** Computer support for access, aggregation, and application is (largely) restricted to the semantic level.
- **This talk:** How do we do maths and math documents at the semantic level?

Hilbert's (Formalist) Program

- **Definition 2.1.** Hilbert's Program called for a foundation of mathematics with
  - A formal system that can express all of mathematics (language, models, calculus)
  - Completeness: all valid mathematical statements can be proved in the formalism.
  - Consistency: a proof that no contradiction can be obtained in the formalism of mathematics.
  - Decidability: algorithm for deciding the truth or falsity of any mathematical statement.
- Originally proposed as "metamathematics" by David Hilbert in 1920.
- **Evaluation:**
  The program was
  - successful in that FOL+ZFC is a foundation [Göd30] (there are others)
  - disappointing for completeness [Göd31], consistency [Göd31], decidability [Chu36; Tur36]
  - inspiring for computer scientists building theorem provers
  - largely irrelevant to current mathematicians (I want to address this!)

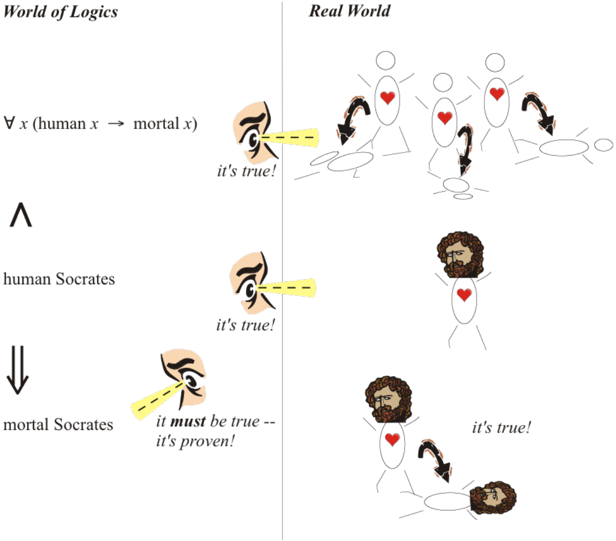
Formality in Logic and Artificial Intelligence

- AI, Philosophy, and Math identify formal representations with Logic
- **Definition 2.2.** A formal system $\mathcal{S}{:=}\langle\mathcal{L},\mathcal{M},\mathcal{C}\rangle$ consists of
  - a (computable) formal language $\mathcal{L}{:=}\mathcal{L}(\mathcal{S})$ (grammar for words/sentences)
  - a model theory $\mathcal{M}$, (a mapping into (some) world)
  - and a sound (complete?) proof calculus $\mathcal{C}$ (a syntactic method of establishing truth)

  We use $\mathfrak{F}$ for the class of all formal systems.
- Reasoning in a formal system proceeds like a chess game: chaining "moves" allowed by the proof calculus via syntactic (depending only on the form) criteria.
- **Observation:** computers need $\mathcal{L}$ and $\mathcal{C}$ (adequacy hinges on relation to $\mathcal{M}$)
- Formality is a "all-or-nothing property". (a single "clearly" can ruin a formal proof)
- **Empirically:** formalization is not always achievable (too tedious for the gain!)
- Humans can draw conclusions from informal (not $\mathcal{L}$) representations by other means (not $\mathcal{C}$).

The miracle of logics

- Purely formal derivations are true in the real world!



World of Logics

$\forall x\,(\text{human}\ x \rightarrow \text{mortal}\ x)$

it's true!

$\wedge$

human Socrates

it's true!

$\Downarrow$

mortal Socrates

*it **must** be true --*
it's proven!

Real World

it's true!

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 2.5.** 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 2.6.** The de Bruijn factor is the quotient of the lengths of the formalization and the original text.

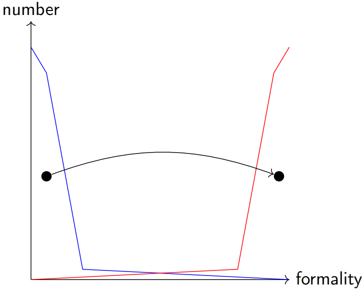
Formalization in Mathematical Practice

- 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 2.7.** 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 2.8.** 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?

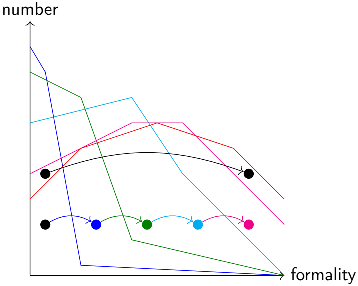
Migration by Stepwise Formalization

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



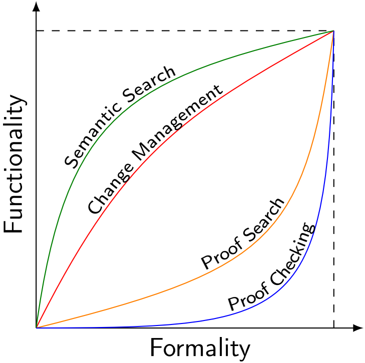
Migration by Stepwise Formalization

- 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 ⇝ 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

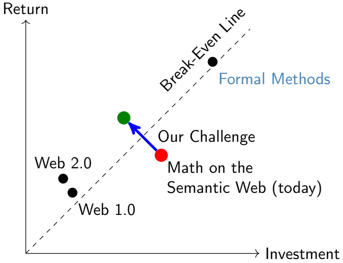


The Flexiformalist Program (Details in [Koh13])

- 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.

Applications!

- A Business model for a Semantic Web for Math/Science?
- For uptake it is essential to match the return to the investment!



- Need to move the technology up (carrots) and left (easier)

2.3 What is formality?

The Process of Formalization

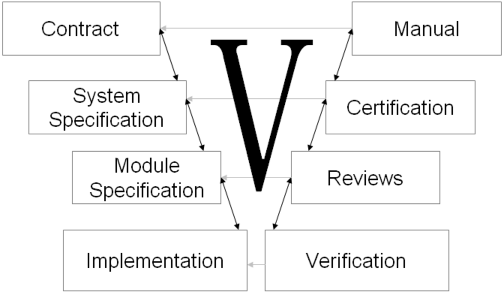
- Formalization in mathematics can be seen as a sequence of documents
  1. an informal proof sketch on a blackboard, and
  2. a high-level run-through of the essentials of a proof in a colloquium talk,
  3. and the speaker's notes that contain all the *detail*s that are glossed over in
  4. a *fully rigorous proof published in a journal*, which may lead to
  5. a *mechanical verification* of the proof in a *proof checker*. (This is formal!)
- Intuitively, the steps get ever more formal, but our definition cannot predict this.
- **Example 3.1.** A recap of concepts from the intro of [CS09]

  An accelerated Turing machine (sometimes called Zeno machine) is a Turing machine that takes 2^{-n} units of time (say seconds) to perform its n^{th} step.
- **Example 3.2.** A rigorous definition of the same concept.

  ***Definition 1.3**: An **accelerated Turing machine** is a Turing machine $M = \langle X, \Gamma, S, s_o, \Box, \delta \rangle$ working with with a computational time structure $T = \langle \{t_i\}_i, <, + \rangle$ with $T \subseteq \mathbb{Q}_+$ ($\mathbb{Q}_+$ is the set of non-negative rationals) such that $\sum_{i \in \mathbb{N}} t_i < \infty$.*

Multiple Dimensions in Formalization I

- **Example 3.3 (SAMS Case Study).** Formalize a set of robot design documents down to implementation and up again to documentation.

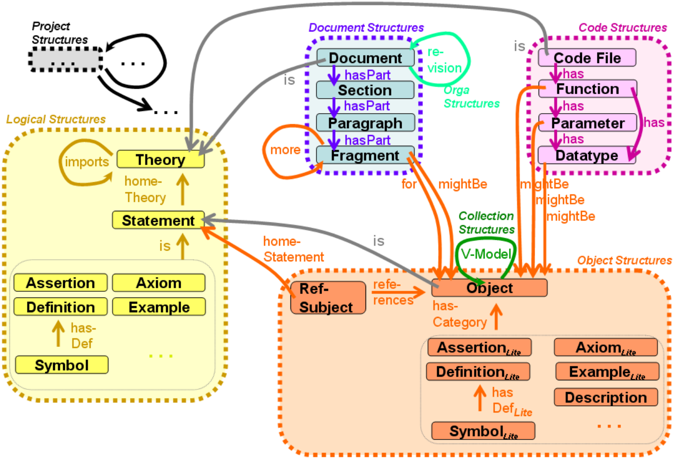


The V-Model requires explicit cross-references between the levels

- **Observation:** The links between the document fragments are formalized by a graph structure for machine support. (e.g. requirements tracing)

Multiple Dimensions in Formalization II

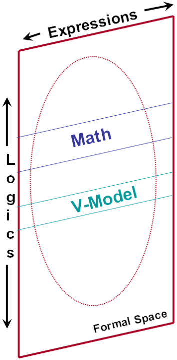
- We ended with a complex, multi-dimensional collection domain model

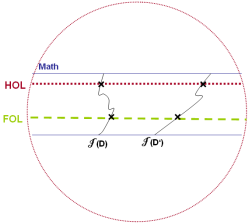


- In particular, the formalization process was linear in the dimensions at best.

What is Informal Mathematical Knowledge

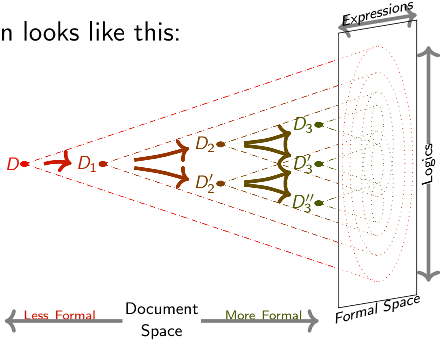
- **Idea:** Informal knowledge could be formalized (but isn't yet!)
- **Definition 3.4.** The meaning of a knowledge item is the set of all its formalizations.
- **Problem:** What is the space of formalizations?
- **Definition 3.5.** The formal space is the set $\mathcal{F}:=\{\langle S,e\rangle|S\in\mathfrak{F},e\in\mathcal{L}(S)\}$, 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 $\mathcal{F}$

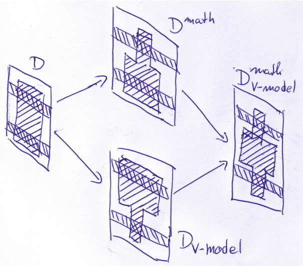
- Stepwise formalization looks like this:



- **Definition 3.6.** $\mathcal{D}$ 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:**


$\mathcal{D}$
$\mathcal{D}^{math}$
$\mathcal{D}^{math}_{V\text{-}model}$
$\mathcal{D}_{V\text{-}model}$

  - 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 3.7.** We call a representation flexiform, iff it is of flexible formality in any of the adequate dimensions of formality.

Flexiforms and Flexiformalization

- **Definition 3.8.** “Flexiform” is an adjective, we are interested in
  - flexiform fragments: e.g. definitions with formulae in MathML parallel markup (presentation/content).
  - flexiform theories: formal theories with flexiform fragments.
  - flexiform digital libraries: formality widely ranging, supports flexiformalization in collection.

  Call all such representations flexiforms (noun)
- **Remark:** The set of flexiforms has very good closure properties.
  - Flexiform fragments can be composed to flexiform documents,
  - which can be collected to flexiform libraries,
  - which in turn can be formalized to flexiform theory graphs
  - or excerpted to flexiform documents.

  All that without leaving the space of flexiforms!

2.4 A “formal” Theory of Flexiformality

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)

OMDoc in a Nutshell (three levels of modeling) [Koh06]

Formula level OpenMath/C-MathML

- Objects as logical formulae
- symbol meaning by reference to theory level

```
<apply>
 <csymbol cd="ring">plus</c.>
 <csymbol cd="ring">zero</c.>
 <ci>N</ci>
</apply>
```

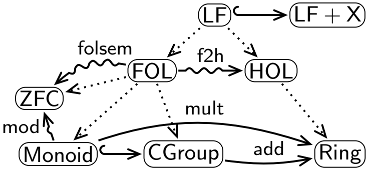
Statement level:

- Definition, Theorem, Proof, Example
- semantics via explicit forms and refs.
- parallel formal & natural language

```
<defn for="plus" type="rec">
 <CMP>rec. eq. for plus</CMP>
 <FMP>X + 0 = X</FMP>
 <FMP>X + s(Y) = s(X + Y)</FMP>
</defn>
```

Module level Theory Graph [RK13]

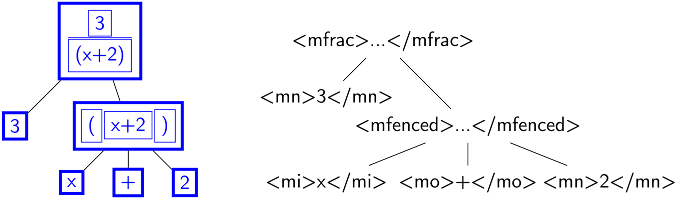
- inheritance via symbol-mapping
- views by proof-obligations
- logics as meta-theories (logic atlas)
- meta-logics as oracles for type/eq



2.4.1 Parallel Markup in MathML

Layout Schemata and the MathML Box model

- Presentation MathML represents the visual appearance of a formula in a tree of layout primitives
- **Example 4.1 (Presentation MathML for $3/(x+2)$).**



<mfrac>...</mfrac>

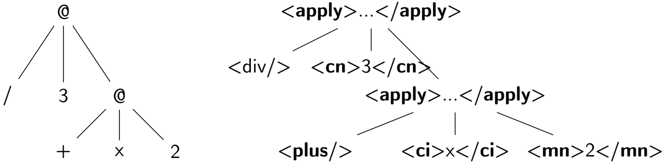
<mn>3</mn>

<mfenced>...</mfenced>

<mi>x</mi> <mo>+</mo> <mn>2</mn>

Functional Markup in MathML: The “Operator Tree”

- Content MathML represents the functional structureof a formula in a tree of operators, via application and binding.
- **Example 4.2 (Content MathML for $3/(x+2)$).**



@

/ 3 @

+ x 2

<**apply**>...</**apply**>

<div/> <**cn**>3</**cn**>

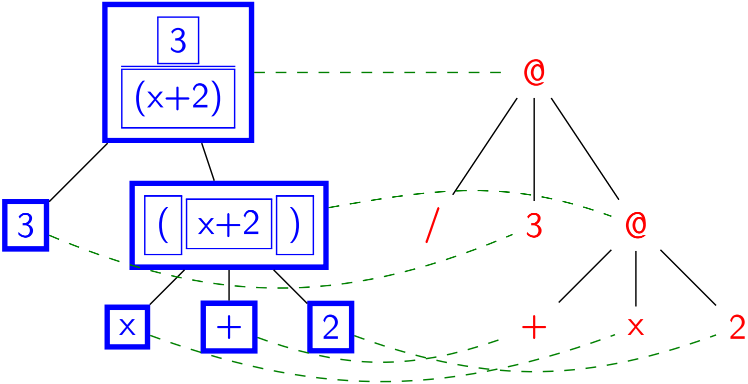
<**apply**>...</**apply**>

<**plus**/> <**ci**>x</**ci**> <**mn**>2</**mn**>

- **Extra Operators:** use <csymbol cd="⟪CD⟫">⟪Name⟫</csymbol>, where
  - CD is a content dictionary a document that defines Name
  - Name is the name of a symbol definition in CD.

Parallel Markup e.g. in MathML I

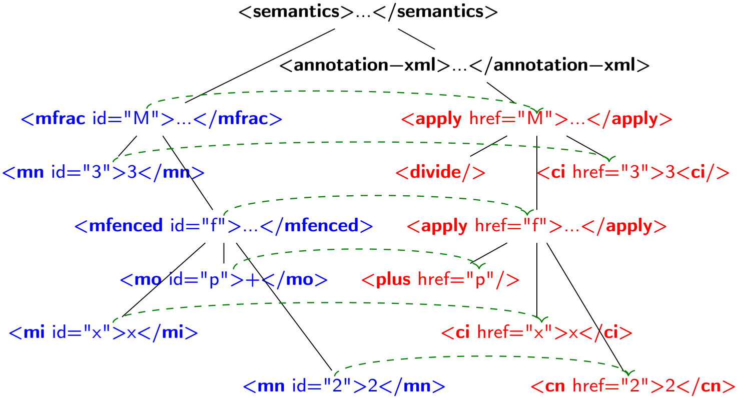
- **Idea:** Combine the presentation and content markup and cross-reference



- 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



<**semantics**>...</**semantics**>

<**annotation−xml**>...</**annotation−xml**>

<**mfrac** id="M">...</**mfrac**>

<**apply** href="M">...</**apply**>

<**mn** id="3">3</**mn**>

<**divide**/>

<**ci** href="3">3<**ci**/>

<**mfenced** id="f">...</**mfenced**>

<**apply** href="f">...</**apply**>

<**mo** id="p">+</**mo**>

<**plus** href="p"/>

<**mi** id="x">x</**mi**>

<**ci** href="x">x</**ci**>

<**mn** id="2">2</**mn**>

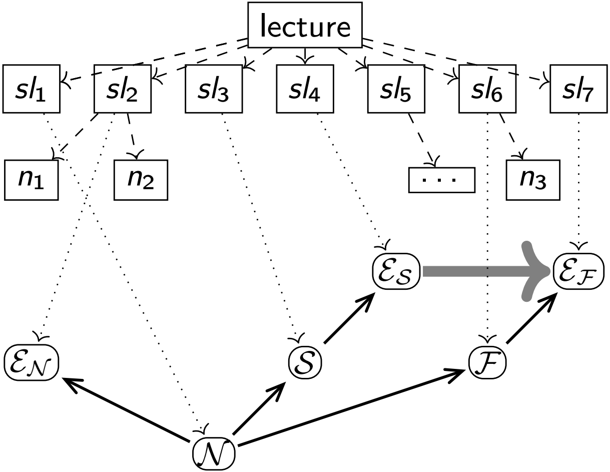
<**cn** href="2">2</**cn**>

x0

2.4.2 Parallel Markup in OMDoc

Separating Narrative– and Conceptual Structure

- Document structure is discourse-level presentation of content structure.
- **Example 4.3.** Introducing a theory via a straw man in a lecture



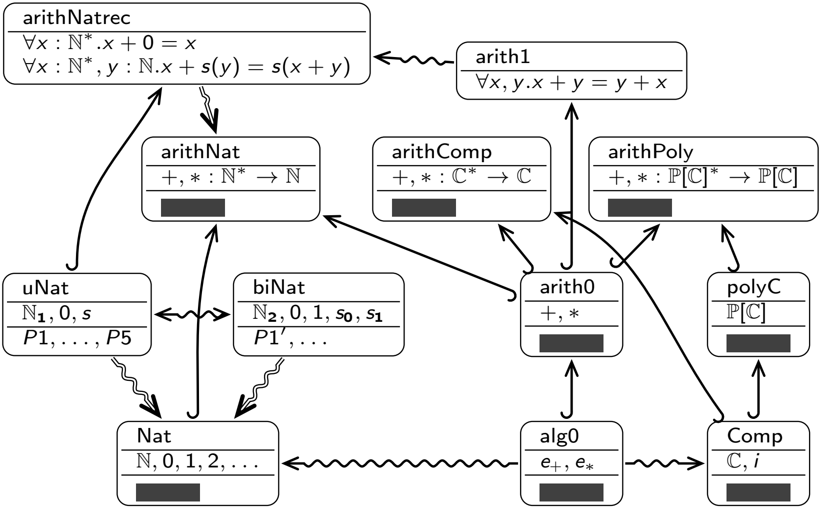
- sl_i are slides
- n_i is narrative text
- $\mathcal{E}_i$ are examples
- $\mathcal{N}$ is a naive theory
- $\mathcal{F}$ is the final theory
- $\mathcal{S}$ is the straw man

- **Idea:** have two documents content + narrative structure
- **Narrative OMDoc:** only doc. structure + narr. elements + links into content.
- **Future:** Generate the narr. from content (need discourse-level content markup)

2.4.3 Flexible Symbol Grounding in OMDoc

A Formal Theory of Underspecification?

- Use theory graphs to specify "meaning" in stages e.g. arithmetics



- **Be non-committal:** In OpenMath, `arith1.ocd` only says that + is commutative

this is a feature, not a bug (lets you remain uncommitted/underspecified)

2.5 Representing Mathematical Vernacular

Chapter 3
Summary and Review

3.1 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)
- **Definition 1.1.** In the theory graph paradigm, we have
  - theories as collections of symbol declarations and axioms (model assumptions)
  - theory morphisms as mappings that translate axioms into theorems

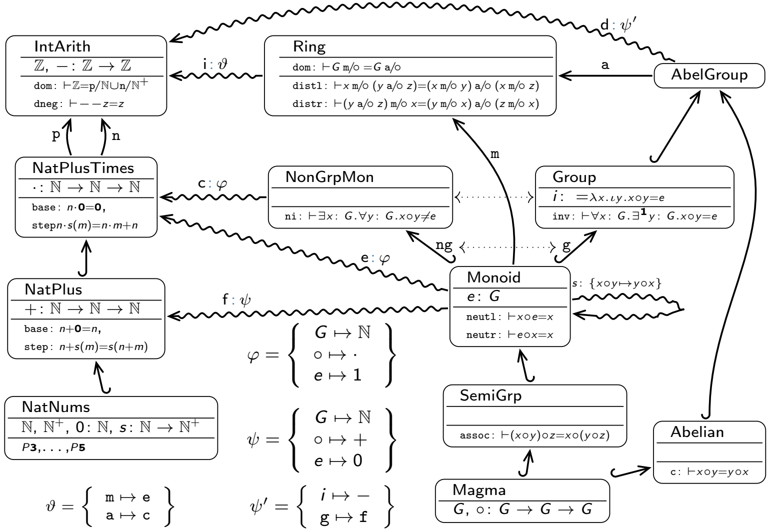
  The central object of knowledge curation is the theory graph which has theories as nodes and theory morphisms as edge.
- **Example 1.2 (MMT: Modular Mathematical Theories).** MMT is a foundation-independent theory graph formalism with advanced theory morphisms.

The Theory Graph Paradigm

- **Definition 1.3.** In the little theories doctrine, theories are made as small as reasonable to enhance modularity and re-use.
- **Definition 1.4.** In the tiny theories doctrine theories are minimal, i.e. have at most two declarations. (one inclusions and one payload)
- **Problem:** With a proliferation of abstract (tiny) theories readability and accessibility suffers (one reason why the Bourbaki books fell out of favor)

Modular Representation of Math (MMT Example)

- **Example 1.5 (Elementary Algebra and Arithmetics).**



IntArith
$\mathbb{Z}$, $-\colon \mathbb{Z}\to\mathbb{Z}$
dom: $\vdash \mathbb{Z}=\mathtt{p}/\mathbb{N}\cup\mathtt{n}/\mathbb{N}^+$
dneg: $\vdash --z=z$
Ring
dom: $\vdash G\ \mathtt{m}/\circ = G\ \mathtt{a}/\circ$
distl: $\vdash x\ \mathtt{m}/\circ\ (y\ \mathtt{a}/\circ\ z)=(x\ \mathtt{m}/\circ\ y)\ \mathtt{a}/\circ\ (x\ \mathtt{m}/\circ\ z)$
distr: $\vdash (y\ \mathtt{a}/\circ\ z)\ \mathtt{m}/\circ\ x=(y\ \mathtt{m}/\circ\ x)\ \mathtt{a}/\circ\ (z\ \mathtt{m}/\circ\ x)$
AbelGroup
d : ψ'
i : ϑ
a
p
n
m
NatPlusTimes
$\cdot\colon \mathbb{N}\to\mathbb{N}\to\mathbb{N}$
base: $n\cdot\mathbf{0}=\mathbf{0}$,
step$n\cdot s(m)=n\cdot m+n$
c : φ
NonGrpMon
ni: $\vdash\exists x\colon G.\forall y\colon G.x\circ y\neq e$
Group
$i\colon\ =\lambda x.\iota y.x\circ y=e$
inv: $\vdash\forall x\colon G.\exists^{\mathbf{1}}y\colon G.x\circ y=e$
ng
g
e : φ
Monoid
$e\colon G$
neutl: $\vdash x\circ e=x$
neutr: $\vdash e\circ x=x$
s: $\{x\circ y\mapsto y\circ x\}$
NatPlus
$+\colon \mathbb{N}\to\mathbb{N}\to\mathbb{N}$
base: $n+\mathbf{0}=n$,
step: $n+s(m)=s(n+m)$
f : ψ
NatNums
$\mathbb{N}$, $\mathbb{N}^+$, $0\colon\mathbb{N}$, $s\colon\mathbb{N}\to\mathbb{N}^+$
$P3,\ldots,P5$
SemiGrp
assoc: $\vdash(x\circ y)\circ z=x\circ(y\circ z)$
Abelian
c: $\vdash x\circ y=y\circ x$
Magma
G, $\circ\colon G\to G\to G$

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

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

$$\vartheta=\left\{\begin{array}{l} \mathtt{m}\mapsto\mathtt{e}\\ \mathtt{a}\mapsto\mathtt{c}\end{array}\right\}$$

$$\psi'=\left\{\begin{array}{l} i\mapsto -\\ \mathtt{g}\mapsto\mathtt{f}\end{array}\right\}$$

The MMT Module System

- **Central notion:** theory graph with theory nodes and theory morphisms as edges
- **Definition 1.6.** In MMT, a theory is a sequence of constant declarations optionally with type declarations and definitions
- MMT employs the Curry/Howard isomorphism and treats
  - axioms/conjectures as typed symbol declarations (propositions-as-types)
  - inference rules as function types (proof transformers)
  - theorems as definitions (proof terms for conjectures)
- **Definition 1.7.** MMT had two kinds of theory morphisms
  - structures instantiate theories in a new context (also called: definitional link, import) they import of theory S into theory T induces theory morphism $S \to T$
  - views translate between existing theories (also called: postulated link, theorem link) views transport theorems from source to target (framing).
- Together, structures and views allow a very high degree of re-use
- **Definition 1.8.** We call a statement t induced in a theory T, iff there is
  - a path of theory morphisms from a theory S to T with (joint) assignment σ,
  - such that $t = \sigma(s)$ for some statement s in S.
- **Definition 1.9.** In MMT, all induced statements have a canonical name, the MMT URI.

Applications for Theories in Physics

- Theory Morphisms allow to “view” source theory in terms of target theory.
- Theory Morphisms occur in Physics all the time.

Theory	Temp. in Kelvin	Temp. in Celsius	Temp. in Fahrenheit
Signature	°K	°C	°F
Axiom:	absolute zero at 0°K	Water freezes at 0°C	cold winter night: 0°F
Axiom:	$\delta(°K1) = \delta(°C1)$	Water boils at 100°C	domestic pig: 100°F
Theorem:	Water freezes at 271.3°K	domestic pig: 38°C	Water boils at 170°F
Theorem:	cold winter night: 240°K	absolute zero at −271.3°C	absolute zero at −460°F

Views: °C $\xrightarrow{+271.3}$ °K, °C $\xrightarrow{-32/2}$ °F, and °F $\xrightarrow{+240/2}$ °K, inverses.

- **Other Examples:** Coordinate Transformations,
- **Application:** Unit Conversion: apply view morphism (flatten) and simplify with UOM. (For new units, just add theories and views.)
- **Application:** MathWebSearch on flattened theory (Explain view path)

3.2 Application: Serious Games

Framing for Problem Solving (The FrameIT Method)

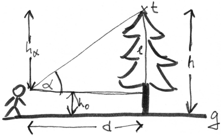
- **Example 2.1 (Problem 0.8.15).**

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

Framing for Problem Solving (The FrameIT Method)

- **Example 2.2 (Problem 0.8.15).**

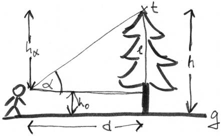
  How can you measure the height of a tree you cannot climb, when you only have a protactor and a tape measure at hand.



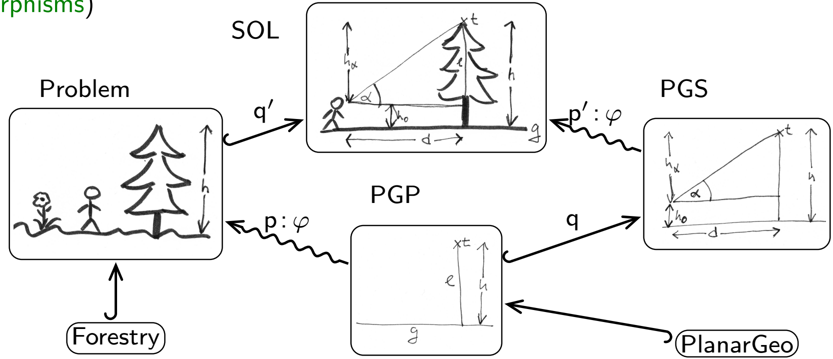
Framing for Problem Solving (The FrameIT Method)

- **Example 2.3 (Problem 0.8.15).**

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



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



SOL

Problem

PGS

PGP

Forestry

PlanarGeo

q'

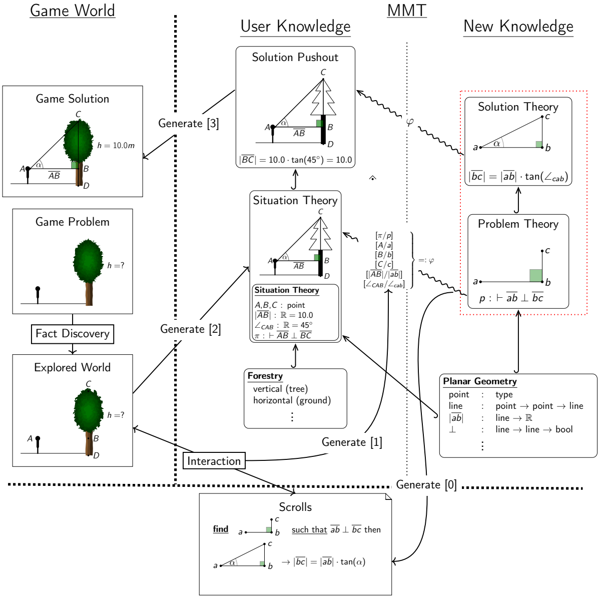
$p' : \varphi$

$p : \varphi$

q

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

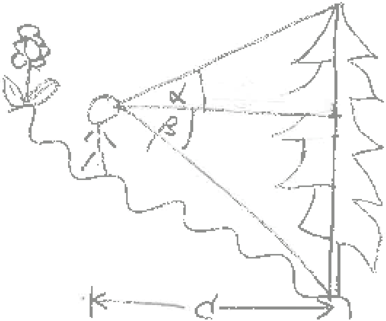
Example Learning Object Graph


Game World
MMT
User Knowledge
New Knowledge
Game Solution
C
$h = 10.0m$
A
α
B
$\overline{AB}$
D
Game Problem
$h = ?$
Fact Discovery
Explored World
C
$h = ?$
A
B
D
Generate [3]
Generate [2]
Generate [1]
Generate [0]
Interaction
Solution Pushout
C
A
α
B
$\overline{AB}$
D
$|\overline{BC}| = 10.0 \cdot \tan(45°) = 10.0$
φ
Situation Theory
C
A
α
B
$\overline{AB}$
D
Situation Theory
A, B, C : point
$|\overline{AB}|$: $\mathbb{R} = 10.0$
$\angle_{CAB}$: $\mathbb{R} = 45°$
π : $\vdash \overline{AB} \perp \overline{BC}$
Forestry
vertical (tree)
horizontal (ground)
⋮
$[\pi/p]$
$[A/a]$
$[B/b]$
$[C/c]$
$[|\overline{AB}|/|\overline{ab}|]$
$[\angle_{CAB}/\angle_{cab}]$
$=: \varphi$
Solution Theory
c
a
α
b
$|\overline{bc}| = |\overline{ab}| \cdot \tan(\angle_{cab})$
Problem Theory
c
a
b
$p : \vdash \overline{ab} \perp \overline{bc}$
Planar Geometry
point : type
line : point → point → line
$|\overline{ab}|$: line → $\mathbb{R}$
$\perp$: line → line → bool
⋮
Scrolls
find
a
b
c
such that $\overline{ab} \perp \overline{bc}$ then
a
α
b
c
$\to |\overline{bc}| = |\overline{ab}| \cdot \tan(\alpha)$

FrameIT Method: Problem

- Problem Representation in the game world (what the student should see)
  Watch
- Student can interact with the environment via gadgets so solve problems
- “Scrolls” of mathematical knowledge give hints.

Combining Problem/Solution Pairs

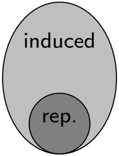


- We can use the same mechanism for combining P/S pairs
- create more complex P/S pairs (e.g. for trees on slopes)

3.3 Search in the Mathematical Knowledge Space

The Mathematical Knowledge Space

- **Observation 3.1.** *The value of framing is that it induces new knowledge*
- **Definition 3.2.** 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

♭search: Indexing flattened Theory Graphs

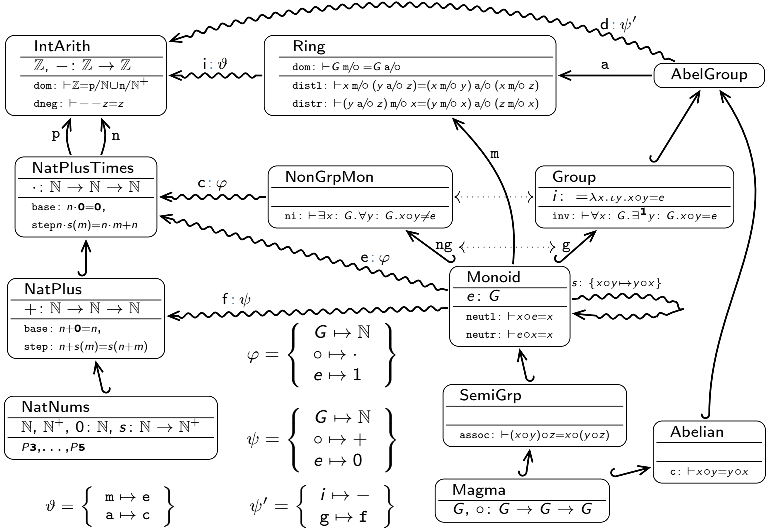
- **Simple Idea:** We have all the necessary components: MMT and MathWebSearch
- **Definition 3.3.** The ♭search systen is an integration of MathWebSearch and MMT that
  - computes the induced formulae of a modular mathematical library via MMT (aka. flattening)
  - indexes induced formulae by their MMT URIs in MathWebSearch
  - uses MathWebSearch for unification-based querying (hits are MMT URIs)
  - uses the MMT to present MMT URI (compute the actual formula)
  - generates explanations from the MMT URI of hits.
- Implemented by Mihnea Iancu in ca. 10 days (MMT harvester pre-existed)
  - almost all work was spent on improvements of MMT flattening
  - MathWebSearch just worked (web service helpful)

♭search User Interface: Explaining MMT URIs

- **Recall:** ♭search (`MathWebSearch` really) returns a MMT URI as a hit.
- **Question:** How to present that to the user? (for his/her greatest benefit)
- **Fortunately:** MMT system can compute induced statements (the hits)
- **Problem:** Hit statement may look considerably different from the induced statement
- **Solution:** Template-based generation of NL explanations from MMT URIs.
  MMT knows the necessary information from the components of the MMT URI.

Modular Representation of Math (MMT Example)

- **Example 3.4 (Elementary Algebra and Arithmetics).**



IntArith
$\mathbb{Z}$, $-\colon \mathbb{Z}\to\mathbb{Z}$
dom: $\vdash \mathbb{Z}=p/\mathbb{N}\cup n/\mathbb{N}^+$
dneg: $\vdash --z=z$

Ring
dom: $\vdash G$ m/∘ $=G$ a/∘
distl: $\vdash x$ m/∘ $(y$ a/∘ $z)=(x$ m/∘ $y)$ a/∘ $(x$ m/∘ $z)$
distr: $\vdash(y$ a/∘ $z)$ m/∘ $x=(y$ m/∘ $x)$ a/∘ $(z$ m/∘ $x)$

AbelGroup

NatPlusTimes
$\cdot\colon \mathbb{N}\to\mathbb{N}\to\mathbb{N}$
base: $n\cdot\mathbf{0}=\mathbf{0}$,
step$n\cdot s(m)=n\cdot m+n$

NonGrpMon
ni: $\vdash\exists x\colon G.\forall y\colon G.x\circ y\neq e$

Group
$i\colon\ =\lambda x.\iota y.x\circ y=e$
inv: $\vdash\forall x\colon G.\exists^{\mathbf{1}}y\colon G.x\circ y=e$

Monoid
$e\colon G$
neutl: $\vdash x\circ e=x$
neutr: $\vdash e\circ x=x$

NatPlus
$+\colon \mathbb{N}\to\mathbb{N}\to\mathbb{N}$
base: $n+\mathbf{0}=n$,
step: $n+s(m)=s(n+m)$

NatNums
$\mathbb{N}$, $\mathbb{N}^+$, $\mathbf{0}\colon\mathbb{N}$, $s\colon\mathbb{N}\to\mathbb{N}^+$
$P\mathbf{3},\ldots,P\mathbf{5}$

SemiGrp
assoc: $\vdash(x\circ y)\circ z=x\circ(y\circ z)$

Abelian
c: $\vdash x\circ y=y\circ x$

Magma
G, $\circ\colon G\to G\to G$

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

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

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

Example: Explaining a MMT URI

- **Example 3.5.** bsearch search result u?IntArith?c/g/assoc for query $(\boxed{x}+\boxed{y})+\boxed{z}=\boxed{R}$.
  - localize the result in the theory u?IntArithf with

    Induced statement $\forall x,y,z : \mathbb{Z}.(x+y)+z = x+(y+z)$ *found in* *http://cds.omdoc.org/cds/elal?IntArith (subst, justification).*
  - Justification: from MMT info about morphism c (source, target, assignment)

    IntArith is a CGroup if we interpret $\circ$ *as* $+$ *and* G *as* $\mathbb{Z}$.
  - skip over g, since its assignment is trivial and generate

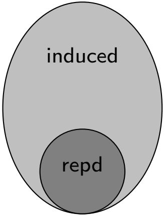
    CGroups are SemiGrps by construction
  - ground the explanation by

    In SemiGrps we have the axiom assoc : $\forall x,y,z : G.(x\circ y)\circ z = x\circ(y\circ z)$

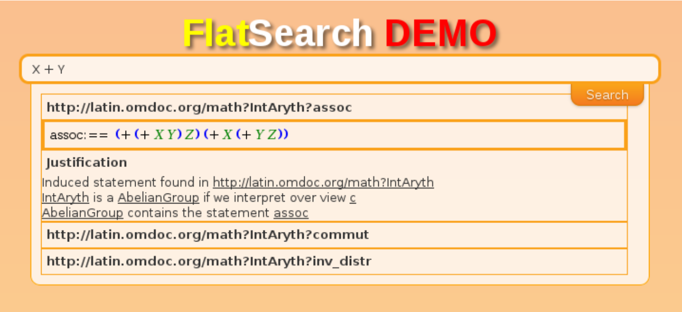
♭search on the LATIN Logic Atlas

- 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`



http://latin.omdoc.org/math?IntAryth?assoc

assoc: == $(+\,(+\,X\;Y)\;Z)\;(+\,X\;(+\,Y\;Z))$

Justification

Induced statement found in http://latin.omdoc.org/math?IntAryth
IntAryth is a AbelianGroup if we interpret over view c
AbelianGroup contains the statement assoc

http://latin.omdoc.org/math?IntAryth?commut

http://latin.omdoc.org/math?IntAryth?inv_distr

Overview: KWARC Research and Projects

Applications: eMath 3.0, Active Documents, Active Learning, Semantic Spreadsheets/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
- Mathematical Models/Simulation

KM & Interaction:

- Semantic Interpretation (aka. Framing)
- math-literate interaction
- MathHub: math archives & active docs
- Active documents: embedded semantic services
- Model-based Education

Semantization:

- LaTeXML: LaTeX → XML
- sTeX: Semantic LaTeX
- invasive editors
- Context-Aware IDEs
- Mathematical Corpora
- Linguistics of Math
- ML for Math Semantics Extraction

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. (Framing as mathematical practice)

References I

[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.

[BC01] 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.

[Bou68] Nicolas Bourbaki. *Theory of Sets*. Elements of Mathematics. Springer Verlag, 1968.

[Bou74] Nicolas Bourbaki. *Algebra I*. Elements of Mathematics. Springer Verlag, 1974.

[Bou89] N. Bourbaki. *General Topology 1-4*. Elements of Mathematics. Springer Verlag, 1989.

References II

[Can95] Georg Cantor. "Beiträge zur Begründung der transfiniten Mengenlehre (1)". In: *Mathematische Annalen* 46 (1895), pp. 481–512. doi: 10.1007/bf02124929.

[Can97] Georg Cantor. "Beiträge zur Begründung der transfiniten Mengenlehre (2)". In: *Mathematische Annalen* 49 (1897), pp. 207–246. doi: doi:10.1007/bf01444205.

[Chu36] Alonzo Church. "A note on the Entscheidungsproblem". In: *Journal of Symbolic Logic* (May 1936), pp. 40–41.

[Chu40] Alonzo Church. "A Formulation of the Simple Theory of Types". In: *Journal of Symbolic Logic* 5 (1940), pp. 56–68.

[CS09] Cris Calude and Ludwig Staiger. *A Note on Accelerated Turing Machines*. CDMTCS Research Report 350. Centre for Discrete Mathematics and Theoretical Computer Science, Auckland University, 2009. url: http://www.cs.auckland.ac.nz/CDMTCS/researchreports/350cris.pdf.

References III

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

[Gen34] Gerhard Gentzen. "Untersuchungen über das logische Schließen I". In: *Mathematische Zeitschrift* 39.2 (1934), pp. 176–210.

[Göd30] Kurt Gödel. "Die Vollständigkeit der Axiome des logischen Funktionenkalküls". In: *Monatshefte für Mathematik und Physik* 37 (1930). English Version in [**Heijenoort67**], pp. 349–360.

[Göd31] Kurt Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: *Monatshefte der Mathematischen Physik* 38 (1931). English Version in [**Heijenoort67**], pp. 173–198.

[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.

References IV

[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: `https://kwarc.info/kohlhase/papers/mkm06cp.pdf`.

[Koh06] Michael Kohlhase. *OMDoc – An open markup format for mathematical documents [Version 1.2]*. LNAI 4180. Springer Verlag, Aug. 2006. url: `http://omdoc.org/pubs/omdoc1.2.pdf`.

[Koh13] 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: `https://kwarc.info/kohlhase/papers/synasc13.pdf`.

References V

[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. Springer Verlag, 2006, pp. 331–342. url: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.453.3917&rep=rep1&type=pdf.

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

[ML86] Sounders Mac Lane. *Mathematics Form and Function*. Springer Verlag, 1986.

References VI

[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`.

[RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: *Information & Computation* 0.230 (2013), pp. 1–54. url: `https://kwarc.info/frabe/Research/mmt.pdf`.

[Tur36] Alan Turing. "On computable numbers, with an application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society, Series 2* 42 (June 1936), pp. 230–265.

References VII

[Wie12] 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/.

[Zer08] Ernst Zermelo. "Untersuchungen über die Grundlagen der Mengenlehre. I.". In: *Mathematische Annalen* 65 (1908), pp. 261–281.