Computerising Mathematical Texts with MathLang

FAIROUZ KAMAREDDINE
MathLang:Project started in 2000 by
Fairouz Kamareddine and J.B. Wells.
A number of positions are available, contact
{*fairouz,jbw*}*@macs.hw.ac.uk*
Ph.D. students in the MathLang team:
M. Maarek (10/2002-6/2007)
K. Retel (11/2004-)
R. Lamar (10/2006-)
C. Zengler (01/2008-01/2009)
Numerous undergraduate and MSc students since 2000

ULTRA Group – Heriot-Watt University, Edinburgh, UK
`http://www.macs.hw.ac.uk/ultra/`

ISIICT 2009, Philadelphia University, Amman, Jordan

Some background

- There are two influencing questions:
  1. What is the relationship between logic and mathematics
  2. What is the relationship between computer science and mathematics.
- Question 1 has been slowly brewing for over 2500 years.
- Question 2, is more recent but is unavoidable since automation and computation can provide tremendous services to mathematics.
- There are also extensive opportunities from combining progress in logic and automation/computerisation not only in mathematics but also in other areas: bio-Informatics, chemistry, music, Natural Language, etc.

Did logic fail for mathematics?

- As far back as the Greeks, we know that logic was influential in the study and development of mathematics.
- Aristotle already knew that for a proposition Φ.
  - If you *give* me a proof of Φ, I can check whether this proof really proves Φ.
  - But, if you ask me to *find* a proof of Φ, I may go on forever trying but without success.
- Aristotle used logic to reason about everything (mathematics, law, farming, medicine,...)
- Euclid's geometry's main feature is the logical deductive style developed for reasoning about mathematics.
- In the 17th century, Leibniz wanted to use logic to prove the existence of God.

Logic and mathematics

In the 19th century, the *need for a more precise* style in mathematics arose, *because controversial results* had appeared in *analysis*.

- 1821: Many of these controversies were solved by the work of Cauchy. E.g., he introduced *a precise definition of convergence* in his *Cours d'Analyse* (A.-L. Cauchy 1897).
- 1872: Due to the more *exact definition of real numbers* given by Dedekind (R. Dedekind 1872), the rules for reasoning with real numbers became even more precise.
- 1895-1897: Cantor began formalizing *set theory* (G. Cantor 1895 and 1897) and made contributions to *number theory*.

Formal systems in the 19th century

- 1889: *Peano* formalized *arithmetic* (G. Peano 1989), but did not treat logic or quantification.
- 1879: *Frege* was not satisfied with the use of *natural language in mathematics*:

> *"...I found the inadequacy of language to be an obstacle; no matter how unwieldy the expressions I was ready to accept, I was less and less able, as the relations became more and more complex, to attain the precision that my purpose required."*
>
> (Begriffsschrift, *Preface*)

Frege therefore presented *Begriffsschrift* (G. Frege 1892), the first formalisation of logic giving logical concepts via symbols rather than natural language.

Formal systems in the 19th century

"[Begriffsschrift's] first purpose is to provide us with the most reliable test of the validity of a chain of inferences and to point out every presupposition that tries to sneak in unnoticed, so that its origin can be investigated."

(Begriffsschrift, *Preface*)

- 1892-1903 Frege's *Grundgesetze der Arithmetik* (G. Frege 1892 and 1903) could handle elementary arithmetic, set theory, logic, and quantification.
- Also in early 1900, a number of questions/problems were posed which were to have a huge impact on logic and computation: (e.g., Hilbert's 23 problems).
- One very important question in the early 1900 was: Can any logical statement have a proof or be disproved.
- More than 30 years later, this question was negatively answered by Turing (Turing machines), Goedel (incompleteness results) and Church (λ-calculus).

And so, the birth of computation machines, and limits of computability

- The first half of the 20th century saw a surge of different formalisms and saw the birth of computers (Turing machines, Von Neumann's machine, etc).
- E.g., the discovery of Russell's paradox was the reason for the invention of the first type theory.
- There was a competition between set/type/category theory as a better foundation for mathematics.
- The second half of the 20th century would see a surge of programming languages and softwares for mathematics.

Can we solve/compute everything?

- Turing answered the question in terms of a *computer*.
  Turing's machines are so powerful: *anything that can ever be computed even on the most powerful computers, can also be computed on a Turing machine.*
- Church invented the λ-calculus, a *language for programming*.
  λ-calculus is so powerful: *anything that can ever be computed can be described in the λ-calculus.*
- Goedel's result meant that no absolute guarantee can be given that many significant branches of mathematics are entirely free of contradictions.
- This meant that: we can compute a very small (countable) amount compared to what we will never be able to compute (uncountable).
- Hilbert's dream was shattered. According to the great historian of Mathematics Ivor Grattan-Guinness, Hilbert behaved coldly towards Goedel.

And so!! different theories, different formalisms

- Translations of Mathematics into logic (Hilbert, Ackermann, Weyl, Russell, Whitehead, Frege, etc.) showed that no logic is fully satisfactory.
- First order logics? Higher order logics? Predicative logics/ impredicative ones?
- There are different set theories: well-founded, non well-founded, with/without foundation axiom/axiom of choice, etc.
- There are different type theories: simple, polymorphic, dependent,etc.
- There are arguments that category theory can serve parts of mathematics better than type theory or set theory.
- And new logics, set/type/category theories are regularly being developed.
- Worst, the ordinary mathematician is not interested in any of this progress.

Common Mathematical Language of mathematicians: CML

+ CML is *expressive*: it has linguistic categories like *proofs* and *theorems*.
+ CML has been refined by intensive use and is rooted in *long traditions*.
+ CML is *approved* by most mathematicians as a communication medium.
+ CML *accommodates many branches* of mathematics, and is adaptable to new ones.

– Since CML is based on natural language, it is *informal* and *ambiguous*.
– CML is *incomplete:* Much is left implicit, appealing to the reader's intuition.
– CML is *poorly organised:* In a CML text, many structural aspects are omitted.
– CML is *automation-unfriendly:* A CML text is a plain text and cannot be easily automated.

A CML-text

From chapter 1, § 2 of E. Landau's *Foundations of Analysis* (Landau 1930, 1951).

Theorem 6 (Commutative Law of Addition)

$$x + y = y + x.$$

Proof Fix y, and let $\mathfrak{M}$ be the set of all x for which the assertion holds.
I) We have

$$y + 1 = y',$$

and furthermore, by the construction in the proof of Theorem 4,

$$1 + y = y',$$

so that

$$1 + y = y + 1$$

and 1 belongs to $\mathfrak{M}$.
II) If x belongs to $\mathfrak{M}$, then

$$x + y = y + x,$$

Therefore

$$(x+y)' = (y+x)' = y+x'.$$

By the construction in the proof of Theorem 4, we have

$$x' + y = (x + y)',$$

hence

$$x' + y = y + x',$$

so that x' belongs to $\mathfrak{M}$. The assertion therefore holds for all x. □

The problem with formal logic

- No logical language is an alternative to CML
  - A logical language does not have *mathematico-linguistic* categories, is *not universal* to all mathematicians, and is *not a good communication medium*.
  - Logical languages make fixed choices (*first versus higher order, predicative versus impredicative, constructive versus classical, types or sets*, etc.). But different parts of mathematics need different choices and there is no universal agreement as to which is the best formalism.
  - A logician reformulates in logic their *formalization* of a mathematical-text as a formal, complete text which is structured considerably *unlike* the original, and is of little use to the *ordinary* mathematician.
  - Mathematicians do not want to use formal logic and have *for centuries* done mathematics without it.
- *So, mathematicians kept to CML.*
- We would like to find an alternative to CML which avoids some of the features of the logical languages which made them unattractive to mathematicians.

What are the options for computerization?

Computers can handle mathematical text at various levels:

- Images of pages may be stored. While useful, this is not a good representation of *language* or *knowledge*.
- Typesetting systems like LaTeX, $\mathrm{T_EX_{MACS}}$, can be used.
- Document representations like OpenMath, OMDoc, MathML, can be used.
- Formal logics used by theorem provers (Coq, Isabelle, Mizar, Isar, etc.) can be used.

We are gradually developing a system named MathLang which we hope will eventually allow building a bridge between the latter 3 levels.

This talk aims at discussing the motivations rather than the details.

The issues with typesetting systems

+ A system like LaTeX, $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$, provides good defaults for visual appearance, while allowing fine control when needed.
+ LaTeX and $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ support commonly needed document structures, while allowing custom structures to be created.
- Unless the mathematician is amazingly disciplined, the *logical structure of symbolic formulas is not represented* at all.
- The *logical structure of mathematics as embedded in natural language text is not represented.* Automated discovery of the semantics of natural language text is still too primitive and requires human oversight.

LaTeX example

draft documents	✓
public documents	✓
computations and proofs	✗

```
\begin{theorem}[Commutative Law of Addition]\label{theorem:6}
$$x+y=y+x.$$
\end {theorem}
\begin{proof}
Fix $y$, and $\mathfrak{M}$ be the set of all $x$ for which
the assertion holds.
\begin{enumerate}
\item We have $$y+1=y',$$ and furthermore, by the construction
in the proof of Theorem \ref{theorem:4}, $$1+y=y',$$
so that $$1+y=y+1$$ and $1$ belongs to $\mathfrak{M}$.
\item If $x$ belongs to $\mathfrak{M}$, then $$x+y=y+x,$$
Therefore $$(x+y)'=(y+x)'=y+x'.$$
By the construction in the proof of
Theorem \ref{theorem:4}, we have $$x'+y=(x+y)',$$
hence $$x'+y=y+x',$$ so that $x'$ belongs to $\mathfrak{M}$.
\end{enumerate}
The assertion therefore holds for all $x$.
\end{proof}
```

The differences of OMDoc

OMDoc attempts to solve some of the difficulties of typesetting systems.

+ Translation to LaTeX (still needed) or MathML can handle visual appearance.
– Precise appearance control must work *through* a translation (difficult!).
+ OMDoc supports commonly needed document structures.
+ The tree structure of symbolic formulas is represented.
– The semantics of symbolic formulas is not represented.
– Type checking symbolic formulas (beyond arity) must be outside OMDoc.
– The logical structure of mathematics as embedded in natural language text is still not represented. There are ways to associate symbolic formulas with natural language text, but no way to check their consistency.

The beginnings of computerized formalization

- In 1967 the famous mathematician de Bruijn began work on logical languages for complete books of mathematics that can be *fully* checked by machine.
- People are prone to error, so if a machine can do proof checking, we expect fewer errors.
- Most mathematicians doubted de Bruijn could achieve success, and computer scientists had no interest at all.
- However, he persevered and built *Automath* (AUTOmated MATHematics).
- Today, there is much interest in many approaches to proof checking for verification of computer hardware and software.
- Many theorem provers have been built to mechanically check mathematics and computer science reasoning (e.g. Isabelle, HOL, Coq, etc.).

Full formalization difficulties: choices

A CML-text is structured differently from a fully formalized text proving the same facts. *Making the latter involves extensive knowledge and many choices:*

- The choice of the *underlying logical system.*
- The choice of *how concepts are implemented* (equational reasoning, equivalences and classes, partial functions, induction, etc.).
- The choice of the *formal foundation*: a type theory (dependent?), a set theory (ZF? FM?), a category theory? etc.
- The choice of the *proof checker*: Automath, Isabelle, Coq, PVS, Mizar, ...

An issue is that one must in general commit to one set of choices.

Full formalization difficulties: informality

Any informal reasoning in a CML-text will cause various problems when fully formalizing it:

- A single (big) step may need to expand into a (series of) syntactic proof expressions. *Very long expressions can replace a clear CML-text.*
- The entire CML-text may need *reformulation* in a fully *complete* syntactic formalism where every detail is spelled out. New details may need to be woven throughout the entire text. The text may need to be *turned inside out.*
- Reasoning may be obscured by *proof tactics*, whose meaning is often *ad hoc* and implementation-dependent.

Regardless, ordinary mathematicians do not find the new text useful.

Coq example

draft documents	✗
public documents	✗
computations and proofs	✓

From Module `Arith.Plus` of Coq standard library (`http://coq.inria.fr/`).

```
Lemma plus_sym: (n,m:nat)(n+m)=(m+n).
Proof.
Intros n m ; Elim n ; Simpl_rew ; Auto with arith.
Intros y H ; Elim (plus_n_Sm m y) ; Simpl_rew ; Auto with
arith.
Qed.
```

MathLang's Goal: Open borders between mathematics, logic and computation

- Ordinary mathematicians *avoid* formal mathematical logic.
- Ordinary mathematicians *avoid* proof checking (via a computer).
- Ordinary mathematicians *may use* a computer for computation: there are over 1 million people who use Mathematica (including linguists, engineers, etc.).
- Mathematicians may also use other computer forms like Maple, LaTeX, etc.
- But we are not interested in only *libraries* or *computation* or *text editing*.
- We want *freedeom of movement* between mathematics, logic and computation.
- At every stage, we must have *the choice* of the level of formalilty and the depth of computation.

Aim for MathLang? (Kamareddine and Wells 2001, 2002)

Can we formalise a mathematical text, avoiding as much as possible the ambiguities of natural language, while still guaranteeing the following four goals?

1. The formalised text looks very much like the original mathematical text (and hence the content of the original mathematical text is respected).
2. The formalised text can be fully manipulated and searched in ways that respect its mathematical structure and meaning.
3. Steps can be made to do computation (via computer algebra systems) and proof checking (via proof checkers) on the formalised text.
4. This formalisation of text is not much harder for the ordinary mathematician than LaTeX. *Full formalization down to a foundation of mathematics is not required*, although allowing and supporting this is one goal.

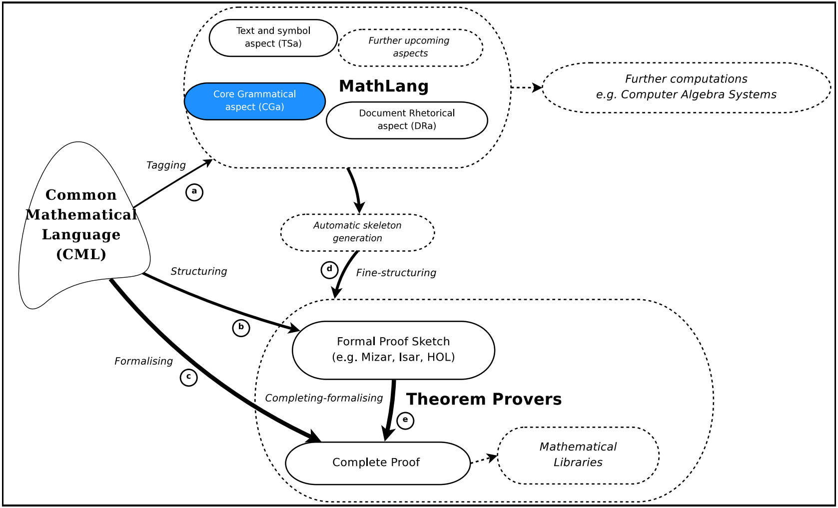
(No theorem prover's language satisfies these goals.)

MathLang

draft documents	✓
public documents	✓
computations and proofs	✓

- A MathLang text captures the grammatical and reasoning aspects of mathematical structure for further computer manipulation.
- A *weak type system* checks MathLang documents at a grammatical level.
- A MathLang text remains *close* to its CML original, allowing confidence that the CML has been captured correctly.
- We have been developing ways to weave natural language text into MathLang.
- MathLang aims to eventually support *all encoding uses*.
- The CML view of a MathLang text should match the mathematician's intentions.
- The formal structure should be suitable for various automated uses.

Example of a MathLang Path
Kamareddine, Maarek, Retel and Wells 2007a
Kamareddine, Wells and Zengler 2008



What is CGa? (Kamareddine, Maarek and Wells 2005)

- CGa is a formal language derived from MV (N.G. de Bruijn 1987) and WTT (Kamareddine and Nederpelt 2004) which aims at expliciting the grammatical role played by the elements of a CML text.
- The structures and common concepts used in CML are captured by CGa with a finite set of grammatical/linguistic/syntactic categories: *Term* "$\sqrt{2}$", *set* "$\mathbb{Q}$", *noun* "number", *adjective* "even", *statement* "$a = b$", *declaration* "Let a be a number", *definition* "An even number is..", *step* "a is odd, hence $a \neq 0$", *context* "Assume a is even".
- Generally, each syntactic category has a corresponding *weak type*.
- CGa's type system Kamareddine, Maarek and Wells 2005 derives typing judgments to check whether the reasoning parts of a document are coherently built.

Examples of linguistic categories

- Terms: the triangle ABC; the center of ABC; $d(x, y)$.
- Nouns: a triangle; an edge of ABC; a group.
- Adjectives: equilateral triangle; prime number; Abelian group.
- Statements: P lies between Q and R; $5 \geq 3$; AB is an edge of ABC.
- Definition: a number p is prime whenever ....

CGa's Commonality with MV

- MV is somewhat faithful to CML yet is formal and avoids ambiguities.
- MV is close to the usual way in which mathematicians write.
- MV has a syntax based on linguistic categories not on set/type theory.
- MV is weak as regards correctness: the rules of MV mostly concern *linguistic* correctness, its types are mostly linguistic so that the formal translation into MV is satisfactory *as a readable, well-organized text*.

Problems with MV

- MV makes many logical and mathematical choices which are best postponed.
- MV incorporates certain correctness requirements, there is for example a hierarchy of types corresponding with sets and subsets.
- MV is already *on its way* to a full formalization, while we want the option of remaining *closer to* a given informal mathematical content.
- A CML text tagged into MathLang
  - has the advantages of the original CML text but not its disadvantages and
  - respects the original CML content.
- *MV does not respect CML content.*

CGa's relation to WTT

- An MV text is not close to its CML original.
- Weak Type Theory, WTT (Kamareddine adn Nederpelt 2004), is MV minus the added logic.
- Although in many ways WTT succeeds and improves on MV, it still fails on respecting the original text. A WTT text is not close to its CML original.
- With CGa, we start from WTT, add some features, and investigate how to integrate it with natural language text.
- Our ongoing development of MathLang is driven by testing it in translating a set of sample texts chosen to cover a large portion of CML usages, both current and historical.
- At the conception of MathLang (Kamareddine and Wells 2001 and 2002) we proposed Euclid's geometry (Heath 1956), Landau's analysis (Landau 1930, 1951), and the Compendium of lattices (Gierz etal 1980) as a start.

Weak Type Theory

In Weak Type Theory (or $\textsc{Wtt}$) we have the following linguistic categories:

- On the *atomic* level: *variables*, *constants* and *binders*,
- On the *phrase* level: *terms* $\mathcal{T}$, *sets* $\mathbb{S}$, *nouns* $\mathcal{N}$ and *adjectives* $\mathcal{A}$,
- On the *sentence* level: *statements* P and *definitions* $\mathcal{D}$,
- On the *discourse* level: *contexts* $\mathbb{I}\Gamma$, *lines* $\mathbf{l}$ and *books* $\mathbf{B}$.

Main categories of syntax of WTT

level	category	abstract syntax	symbol
atomic	*variables*	$\mathtt{V} = \mathtt{V}^T \vert \mathtt{V}^S \vert \mathtt{V}^P$	x
	constants	$\mathtt{C} = \mathtt{C}^T \vert \mathtt{C}^S \vert \mathtt{C}^N \vert \mathtt{C}^A \vert \mathtt{C}^P$	c
	binders	$\mathtt{B} = \mathtt{B}^T \vert \mathtt{B}^S \vert \mathtt{B}^N \vert \mathtt{B}^A \vert \mathtt{B}^P$	b
phrase	*terms*	$\mathcal{T} = \mathtt{C}^T(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^T_{\mathcal{Z}}(\mathcal{E}) \vert \mathtt{V}^T$	t
	sets	$\mathbb{S} = \mathtt{C}^S(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^S_{\mathcal{Z}}(\mathcal{E}) \vert \mathtt{V}^S$	s
	nouns	$\mathcal{N} = \mathtt{C}^N(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^N_{\mathcal{Z}}(\mathcal{E}) \vert \mathcal{A}\mathcal{N}$	n
	adjectives	$\mathcal{A} = \mathtt{C}^A(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^A_{\mathcal{Z}}(\mathcal{E})$	a
sentence	*statements*	$\mathcal{P} = \mathtt{C}^P(\overrightarrow{\mathcal{P}}) \vert \mathtt{B}^P_{\mathcal{Z}}(\mathcal{E}) \vert \mathtt{V}^P$	S
	definitions	$\mathcal{D} = \mathcal{D}^\varphi \vert \mathcal{D}^P$	D
		$\mathcal{D}^\varphi = \mathtt{C}^T(\overrightarrow{V}) := \mathcal{T} \vert \mathtt{C}^S(\overrightarrow{V}) := \mathbb{S} \vert$ $\mathtt{C}^N(\overrightarrow{V}) := \mathcal{N} \vert \mathtt{C}^A(\overrightarrow{V}) := \mathcal{A}$	
		$\mathcal{D}^P = \mathtt{C}^P(\overrightarrow{V}) := \mathcal{P}$	
discourse	*contexts*	$\mathbf{\Gamma} = \emptyset \mid \mathbf{\Gamma}, \mathcal{Z} \mid \mathbf{\Gamma}, \mathcal{P}$	Γ
	lines	$\mathbf{l} = \mathbf{\Gamma} \triangleright \mathcal{P} \mid \mathbf{\Gamma} \triangleright \mathcal{D}$	l
	books	$\mathbf{B} = \emptyset \mid \mathbf{B} \circ \mathbf{l}$	B

Categories of syntax of WTT

Other category	abstract syntax	symbol
expressions	$\mathcal{E} = T \vert \mathbb{S} \vert \mathcal{N} \vert P$	E
parameters	$\mathcal{P} = T \vert \mathbb{S} \vert P$ (note: $\overrightarrow{\mathcal{P}}$ is a list of $\mathcal{P}$s)	P
typings	$\mathbf{T} = \mathbb{S} :$ SET $\vert \mathcal{S} :$ STAT $\vert T : \mathbb{S} \vert T : \mathcal{N} \vert T : \mathcal{A}$	T
declarations	$\mathcal{Z} = \mathtt{V}^S :$ SET $\vert \mathtt{V}^P :$ STAT $\vert \mathtt{V}^T : \mathbb{S} \vert \mathtt{V}^T : \mathcal{N}$	Z

Derivation rules

(1) B is a weakly well-typed book: $\vdash\ B :: \mathbf{B}$.

(2) Γ is a weakly well-typed context relative to book B: $B\ \vdash\ \Gamma :: \mathbb{I}\Gamma$.

(3) t is a weakly well-typed term, etc., relative to book B and context Γ:

$$B;\Gamma\ \vdash\ t :: T, \qquad B;\Gamma\ \vdash\ s :: S, \qquad B;\Gamma\ \vdash\ n :: N,$$
$$B;\Gamma\ \vdash\ a :: A, \qquad B;\Gamma\ \vdash\ p :: P, \qquad B;\Gamma\ \vdash\ d :: D$$

$OK(B;\Gamma)$. stands for: $\vdash\ B :: \mathbf{B}$, *and* $B\ \vdash\ \Gamma :: \mathbb{I}\Gamma$

Examples of derivation rules

- $\texttt{dvar}(\emptyset) = \emptyset \qquad \texttt{dvar}(\Gamma', x : W) = \texttt{dvar}(\Gamma'), x$
  $\texttt{dvar}(\Gamma', P) = \texttt{dvar}(\Gamma')$

$$\frac{OK(B;\Gamma), \quad x \in \texttt{V}^{\texttt{T/S/P}}, \quad x \in \texttt{dvar}(\Gamma)}{B;\Gamma \;\vdash\; x :: T/S/P} \quad (var)$$

$$\frac{B;\Gamma \;\vdash\; n :: N, \quad B;\Gamma \;\vdash\; a :: A}{B;\Gamma \;\vdash\; an :: N} \quad (adj\text{−}noun)$$

$$\frac{}{\vdash\; \emptyset :: \mathbf{B}} \quad (emp\text{−}book)$$

$$\frac{B;\Gamma \;\vdash\; p :: P}{\vdash\; B \circ \Gamma \triangleright p :: \mathbf{B}} \qquad \frac{B;\Gamma \;\vdash\; d :: D}{\vdash\; B \circ \Gamma \triangleright d :: \mathbf{B}}$$

$(book\text{−}ext)$

Properties of MathLang

- *Every variable is declared* If $B; \Gamma \vdash \Phi :: \mathbf{W}$ then $FV(\Phi) \subseteq \mathtt{dvar}(\Gamma)$.
- *Correct subcontexts* If $B \vdash \Gamma :: \mathbb{I}$ and $\Gamma' \subseteq \Gamma$ then $B \vdash \Gamma' :: \mathbb{I}$.
- *Correct subbooks* If $\vdash B :: \mathbf{B}$ and $B' \subseteq B$ then $\vdash B' :: \mathbf{B}$.
- *Free constants are either declared in book or in contexts* If $B; \Gamma \vdash \Phi :: \mathbf{W}$, then $FC(\Phi) \subseteq \mathtt{prefcons}(B) \cup \mathtt{defcons}(B)$.
- *Types are unique* If $B; \Gamma \vdash A :: \mathbf{W_1}$ and $B; \Gamma \vdash A :: \mathbf{W_2}$, then $\mathbf{W_1} \equiv \mathbf{W_2}$.
- *Weak type checking is decidable* there is a decision procedure for the question $B; \Gamma \vdash \Phi :: \mathbf{W}$?.
- *Weak typability is computable* there is a procedure deciding whether an answer exists for $B; \Gamma \vdash \Phi ::$? and if so, delivering the answer.

Definition unfolding

- Let $\vdash B :: \mathbf{B}$ and $\Gamma \triangleright c(x_1, \ldots, x_n) := \Phi$ a line in B.
- We write $B \vdash c(P_1, \ldots, P_n) \xrightarrow{\delta} \Phi[x_i := P_i]$.
- *Church-Rosser* If $B \vdash \Phi \overset{\delta}{\twoheadrightarrow} \Phi_1$ and $B \vdash \Phi \overset{\delta}{\twoheadrightarrow} \Phi_2$ then there exists Φ_3 such that $B \vdash \Phi_1 \overset{\delta}{\twoheadrightarrow} \Phi_3$ andf $B \vdash \Phi_2 \overset{\delta}{\twoheadrightarrow} \Phi_3$.
- *Strong Normalisation* Let $\vdash B :: \mathbf{B}$. For all subformulas Ψ occurring in B, relation $\xrightarrow{\delta}$ is strongly normalizing (i.e., definition unfolding inside a well-typed book is a well-founded procedure).

CGa's grammatical categories (taken from MV/WTT)

term	"$a + b$"
set	"$\mathbb{N}$"
noun	"ring"
adjective	"Abelian"
statement	"$a + 0 = a$"
declaration	"Let a be ..."
definition	"A ring is ..."
step	"..., therefore ..."
context	"Assume ..."

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

- 0 is being declared,

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

- 0 is being declared,
- . . . and is an element of the set R,

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

- 0 is being declared,
- . . . and is an element of the set R,
- a and 0 are terms,

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

- 0 is being declared,
- . . . and is an element of the set R,
- a and 0 are terms,
- Their sum is also a term,

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

- 0 is being declared,
- . . . and is an element of the set R,
- a and 0 are terms,
- Their sum is also a term,
- The equality between $a + 0$ and a is a statement,

Box annotations (categories are CGa, interface is TSa)

There is an element 0 in R such that $a + 0 = a$.

- 0 is being declared,
- ...and is an element of the set R,
- a and 0 are terms,
- Their sum is also a term,
- The equality between $a + 0$ and a is a statement,
- Finally, the overall sentence is a step.

Another example

There is an element $-a$ in R such that $a + (-a) = 0$ for all a in R.

Another example

There is an element $-$a in R such that
a $+ (-$a$) = 0$ for all a in R.

Another example

There is an element $-a$ in R such that $a + (-a) = 0$ for all a in R.

Another example

There is an element $-$a in R such that
a $+ (-$a$) =$ 0 for all a in R.

Another example

There is an element -a in R such that
a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that
a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that
a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that a + (-a) = 0 for all a in R.

Another example

There is an element -a in R such that
a + (-a) = 0 for all a in R.

CGa typing rules

- The CGa syntax is an adaptation of that of WTT and has almost the same categories to both MV and WTT.
- A CGa text can be type checked using CGa type rules which are again an adaptation of those of WTT.
- The automatic type checker type checks a CGa annotated text and if it succeeds, the text is said to be syntactically correct, else a type error message is printed.

CGa Weak Type Checking

T Terms S Sets N Nouns P Statements Z Declarations Γ Context

Let $\mathfrak{M}$ be a set,
y and x are natural numbers,
if x belongs to $\mathfrak{M}$
then $x + y = y + x$

CGa Weak Type checking detects grammatical errors

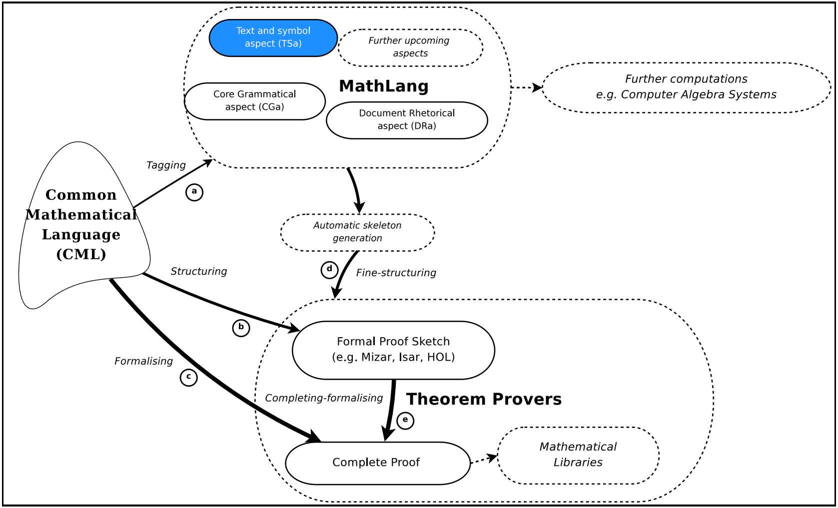
T Terms S Sets N Nouns P Statements Z Declarations Γ Context

Let $\mathfrak{M}$ be a set,
y and x are natural numbers,
if x belongs to $\mathfrak{M}$
then $x + y$ ⇐ **error**

How complete is the CGa?

- CGa is quite advanced but remains under development according to new translations of mathematical texts. Are the current CGa categories sufficient?
- The metatheory of WTT has been established in (Kamareddine and Nederepelt 2004). That of CGa remains to be established. However, since CGa is quite similar to WTT, its metatheory might be similar to that of WTT.
- The type checker for CGa works well and gives some useful error messages. Error messages should be improved.

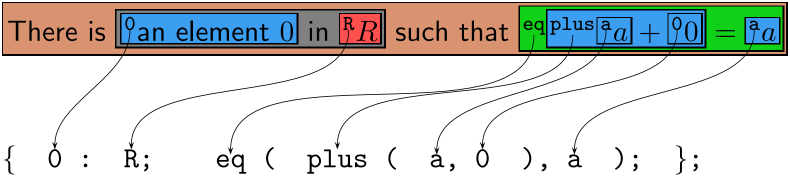
Example of a MathLang Path
Kamareddine, Maarek, Retel and Wells 2007a
Kamareddine, Wells and Zengler 2008



What is TSa? (Kamareddine, Lamar, Maarek and Wells 2007)

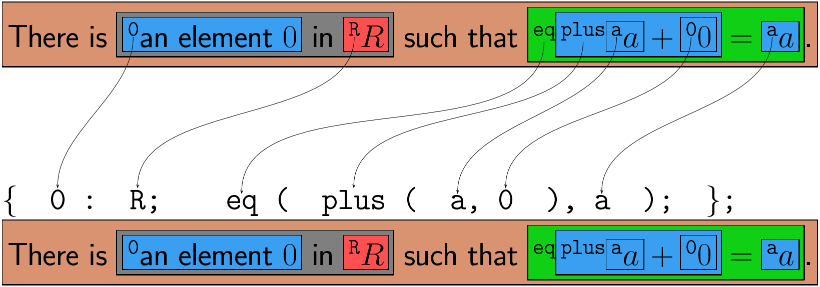
- TSa (Kamareddine, Lamar, Maarek and Wells) builds the bridge between a CML text and its grammatical interpretation and adjoins to each CGa expression a string of words and/or symbols which aims to act as its CML representation.
- TSa plays the role of a user interface
- TSa can flexibly represent natural language mathematics.
- The author wraps the natural language text with boxes representing the grammatical categories (as we saw before).
- The author can also give interpretations to the parts of the text.

Interpretations



At the lower CGa level, these interpretations are helpful for example for dealing with the natural language aspect. At the higher aspects (e.g., filling incomplete proofs), these interpretations could enable assiging intended logical meanings to parts of the text.

Interpretations



There is 0 an element 0 in R R such that eq plus a a + 0 0 equals a a.

0 $0 \in$ R R, eq plus a a + 0 0 $=$ a a.

Rewrite rules enable natural language representation

$$0 + a0 = a0 = a(0 + 0) = a0 + a0$$

eq $0 + a0$ = shared $a0$ eq = shared $a(0+0)$ eq = $a0 + a0$

How do you do this?

$$0+a0 = \qquad a0 = \qquad a(0+0) = a0+a0$$

How do you do this?

$$0+a0 = \quad a0 = \quad a(0+0) = a0+a0$$

How do you do this?

0+a0 =

a0 =

a(0+0) = a0+a0

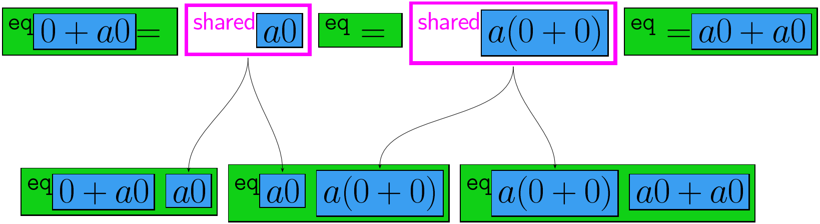
How do you do this?

0+a0 = <share>a0 = <share>a(0+0) = a0+a0

How do you do this?

$0 + a0 =$ <share> $a0$ $=$ <share> $a(0+0)$ $= a0 + a0$

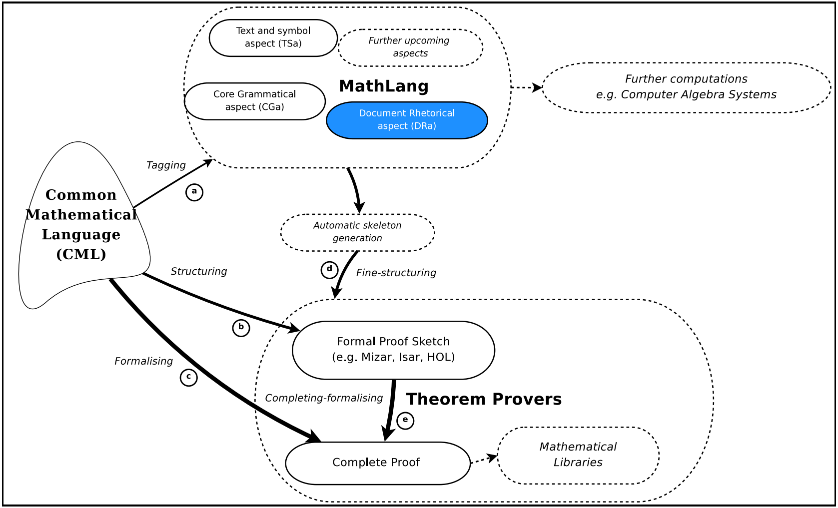
$0 + a0 = a0$ $\quad$ $a0 = a(0+0)$ $\quad$ $a(0+0) = a0 + a0$


eq $0 + a0 =$
shared $a0$
eq $=$
shared $a(0+0)$
eq $= a0 + a0$
eq $0 + a0$ $a0$
eq $a0$ $a(0+0)$
eq $a(0+0)$ $a0 + a0$

How complete is TSa?

- TSa provides useful interface facilities but it is still under development.
- So far, only simple rewrite (souring) rules are used and they are not comprehensive. E.g., unable to cope with things like $\overbrace{x = \ldots = x}^{n \text{ times}}$.
- The TSa theory and metatheory need development.

Example of a MathLang Path
Kamareddine, Maarek, Retel and Wells 2007a
Kamareddine, Wells and Zengler 2008



What is DRa? (Kamareddine, Maarek, Retel and Wells 2007b)

- DRa (Kamareddine, Maarek, Retel and Wells 2007b): Document Rhetorical structure aspect.
- **Structural components of a document** like *chapter, section, subsection, etc.*
- **Mathematical components of a document** like *theorem, corollary, definition, proof, etc.*
- **Relations** between above components.
- These enhance readability, and ease the navigation of a document.
- Also, these help to go into more formal versions of the document.

Relations

Description
Instances of the StructuralRhetoricalRole class: preamble, part, chapter, section, paragraph, *etc.*
Instances of the MathematicalRhetoricalRole class: lemma, corollary, theorem, conjecture, definition, axiom, claim, proposition, assertion, proof, exercise, example, problem, solution, *etc.*

Relation
Types of relations: relatesTo, uses, justifies, subpartOf, inconsistentWith, exemplifies

What does the mathematician do?

- The mathematician wraps into boxes and uniquely names chunks of text
- The mathematician assigns to each box the structural and/or mathematical rhetorical roles
- The mathematician indicates the relations between wrapped chunks of texts

Lemma 1. For $m, n \in \mathbb{N}$ one has: $m^2 = 2n^2 \implies m = n = 0$.
Define on $\mathbb{N}$ the predicate:

$$P(m) \iff \exists n. m^2 = 2n^2 \ \& \ m > 0.$$

Claim. $P(m) \implies \exists m' < m. P(m')$. Indeed suppose $m^2 = 2n^2$ and $m > 0$. It follows that m^2 is even, but then m must be even, as odds square to odds. So $m = 2k$ and we have

$$2n^2 = m^2 = 4k^2 \implies n^2 = 2k^2$$

Since $m > 0$, if follows that $m^2 > 0, n^2 > 0$ and $n > 0$. Therefore $P(n)$. Moreover, $m^2 = n^2 + n^2 > n^2$, so $m^2 > n^2$ and hence $m > n$. So we can take $m' = n$.

By the claim $\forall m \in \mathbb{N}. \neg P(m)$, since there are no infinite descending sequences of natural numbers.

Now suppose $m^2 = 2n^2$ with $m \neq 0$. Then $m > 0$ and hence $P(m)$. Contradiction. Therefore $m = 0$. But then also $n = 0$.

Corollary 1. $\sqrt{2} \notin \mathbb{Q}$.
Suppose $\sqrt{2} \in \mathbb{Q}$, i.e. $\sqrt{2} = p/q$ with $p \in \mathbb{Z}, q \in \mathbb{Z} - \{0\}$. Then $\sqrt{2} = m/n$ with $m = |p|, n = |q| \neq 0$. It follows that $m^2 = 2n^2$. But then $n = 0$ by the lemma. Contradiction shows that $\sqrt{2} \notin \mathbb{Q}$.

Barendregt

Lemma 1. For $m, n \in \mathbb{N}$ one has: $m^2 = 2n^2 \implies m = n = 0$.
Define on $\mathbb{N}$ the predicate:

$$P(m) \iff \exists n. m^2 = 2n^2 \ \& \ m > 0.$$

Claim. $P(m) \implies \exists m' < m. P(m')$. Indeed suppose $m^2 = 2n^2$ and $m > 0$. It follows that m^2 is even, but then m must be even, as odds square to odds. So $m = 2k$ and we have

$$2n^2 = m^2 = 4k^2 \implies n^2 = 2k^2$$

Since $m > 0$, if follows that $m^2 > 0, n^2 > 0$ and $n > 0$. Therefore $P(n)$. Moreover, $m^2 = n^2 + n^2 > n^2$, so $m^2 > n^2$ and hence $m > n$. So we can take $m' = n$.

By the claim $\forall m \in \mathbb{N}. \neg P(m)$, since there are no infinite descending sequences of natural numbers.

Now suppose $m^2 = 2n^2$ with $m \neq 0$. Then $m > 0$ and hence $P(m)$. Contradiction. Therefore $m = 0$. But then also $n = 0$.

Corollary 1. $\sqrt{2} \notin \mathbb{Q}$.
Suppose $\sqrt{2} \in \mathbb{Q}$, i.e. $\sqrt{2} = p/q$ with $p \in \mathbb{Z}, q \in \mathbb{Z} - \{0\}$. Then $\sqrt{2} = m/n$ with $m = |p|, n = |q| \neq 0$. It follows that $m^2 = 2n^2$. But then $n = 0$ by the lemma. Contradiction shows that $\sqrt{2} \notin \mathbb{Q}$.

Barendregt

Lemma 1.

For $m, n \in \mathbb{N}$ *one has:* $m^2 = 2n^2 \implies m = n = 0$ **A**

Proof.

Define on $\mathbb{N}$ the predicate: **E**

$$P(m) \iff \exists n. m^2 = 2n^2 \;\&\; m > 0.$$

Claim. $P(m) \implies \exists m' < m. P(m')$. **F**

Indeed suppose $m^2 = 2n^2$ and $m > 0$. It follows that m^2 is even, but then m must be even, as odds square to odds. **G** So $m = 2k$ and we have $2n^2 = m^2 = 4k^2 \implies n^2 = 2k^2$ Since $m > 0$, if follows that $m^2 > 0$, $n^2 > 0$ and $n > 0$. Therefore $P(n)$. Moreover, $m^2 = n^2 + n^2 > n^2$, so $m^2 > n^2$ **B** and hence $m > n$. So we can take $m' = n$.

By the claim $\forall m \in \mathbb{N}. \neg P(m)$, since there are no infinite descending sequences of natural numbers.

Now suppose $m^2 = 2n^2$

with $m \neq 0$. Then $m > 0$ and hence $P(m)$. **H** Contradiction.

Therefore $m = 0$. But then also $n = 0$. **I** □

Corollary 1. $\sqrt{2} \notin \mathbb{Q}$ **C**

Proof. Suppose $\sqrt{2} \in \mathbb{Q}$, i.e. $\sqrt{2} = p/q$ with $p \in \mathbb{Z}, q \in \mathbb{Z} - \{0\}$. Then $\sqrt{2} = m/n$ with $m = |p|, n = |q| \neq 0$. **D** It follows that $m^2 = 2n^2$. But then $n = 0$ by the lemma. Contradiction shows that $\sqrt{2} \notin \mathbb{Q}$. □

$(A$, hasMathematicalRhetoricalRole, $lemma)$
$(E$, hasMathematicalRhetoricalRole, $definition)$
$(F$, hasMathematicalRhetoricalRole, $claim)$
$(G$, hasMathematicalRhetoricalRole, $proof)$
$(B$, hasMathematicalRhetoricalRole, $proof)$
$(H$, hasOtherMathematicalRhetoricalRole, $case)$
$(I$, hasOtherMathematicalRhetoricalRole, $case)$
$(C$, hasMathematicalRhetoricalRole, $corollary)$
$(D$, hasMathematicalRhetoricalRole, $proof)$

$(B$, justifies, $A)$
$(D$, justifies, $C)$
$(D$, uses, $A)$
$(G$, uses, $E)$
$(F$, uses, $E)$
$(H$, uses, $E)$
$(H$, subpartOf, $B)$
$(H$, subpartOf, $I)$

Lemma 1.

For $m, n \in \mathbb{N}$ *one has:* $m^2 = 2n^2 \implies m = n = 0$

Proof. Define on $\mathbb{N}$ the predicate:

$$P(m) \iff \exists n. m^2 = 2n^2 \;\&\; m > 0.$$

Claim. $P(m) \implies \exists m' < m. P(m')$.

Indeed suppose $m^2 = 2n^2$ and $m > 0$. It follows that m^2 is even, but then m must be even, as odds square to odds. So $m = 2k$ and we have $2n^2 = m^2 = 4k^2 \implies n^2 = 2k^2$ Since $m > 0$, if follows that $m^2 > 0$, $n^2 > 0$ and $n > 0$. Therefore $P(n)$. Moreover, $m^2 = n^2 + n^2 > n^2$, so $m^2 > n^2$ and hence $m > n$. So we can take $m' = n$.

By the claim $\forall m \in \mathbb{N}. \neg P(m)$, since there are no infinite descending sequences of natural numbers.

Now suppose $m^2 = 2n^2$ with $m \neq 0$. Then $m > 0$ and hence $P(m)$. Contradiction.

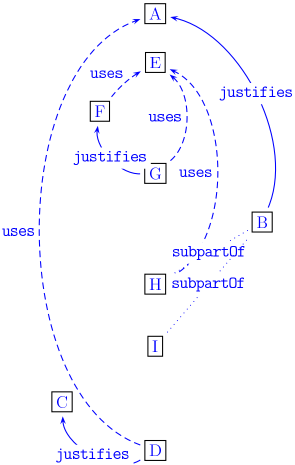
Therefore $m = 0$. But then also $n = 0$. □

Corollary 1. $\sqrt{2} \notin \mathbb{Q}$

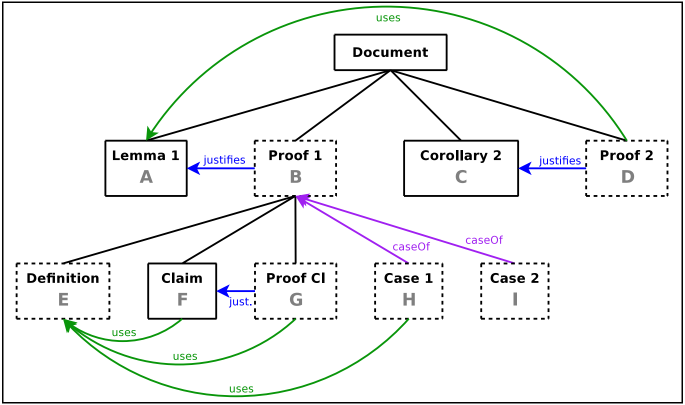
Proof. Suppose $\sqrt{2} \in \mathbb{Q}$, i.e. $\sqrt{2} = p/q$ with $p \in \mathbb{Z}, q \in \mathbb{Z} - \{0\}$. Then $\sqrt{2} = m/n$ with $m = |p|, n = |q| \neq 0$. It follows that $m^2 = 2n^2$. But then $n = 0$ by the lemma. Contradiction shows that $\sqrt{2} \notin \mathbb{Q}$. □

The automatically generated dependency Graph

Dependency Graph (DG)



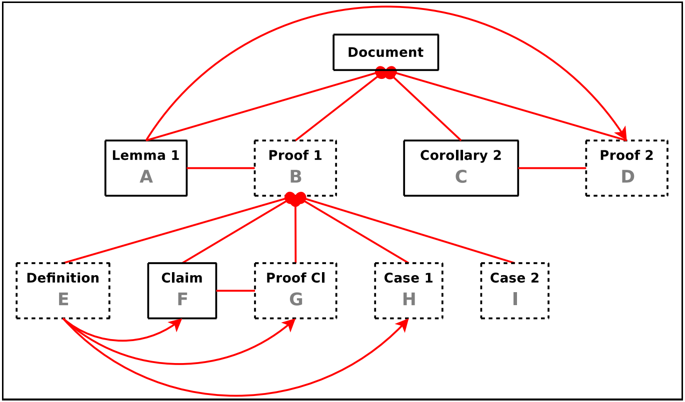
An alternative view of the DRa



The Graph of Textual Order: GoTO
Kamareddine, Wells and Zengler 2008

- To be able to examine the proper structure of a DRa tree we introduce the concept of textual order between two nodes in the tree.
- Using textual orders, we can transform the dependency graph into a GoTO by transforming each edge of the DG.
- So far there are two reasons why the GoTO is produced:
  1. Automatic Checking of the GoTO can reveal errors in the document (e.g. loops in the structure of the document).
  2. The GoTO is used to automatically produce a proof skeleton for a certain prover.
- We automatically transform a DG into GoTO and automatically check the GoTO for errors in the document:
  1. Loops in the GoTO (error)
  2. Proof of an unproved node (error)
  3. More than one proof for a proved node (warning)
  4. Missing proof for a proved node (warning)

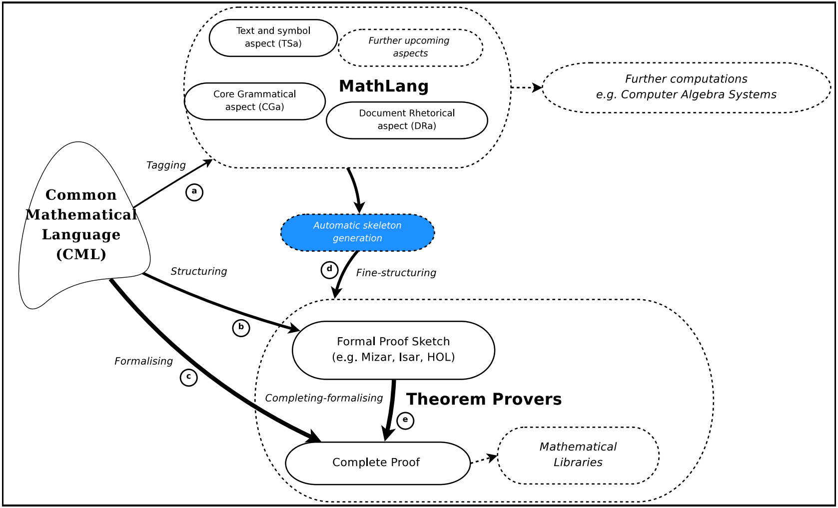
Graph of Textual Order for the DRa tree example



How complete is DRa?

- The dependency graph can be used to check whether the logical reasoning of the text is coherent and consistent (e.g., no loops in the reasoning).
- However, both the DRa language and its implementation need more experience driven tests on natural language texts.
- Also, the DRa aspect still needs a number of implementation improvements (the automation of the analysis of the text based on its DRa features).
- Extend TSa to also cover DRa (in addition to CGa).
- Extend DRa depending on further experience driven translations.
- Establish the soundness and completeness of DRa for mathematical texts.

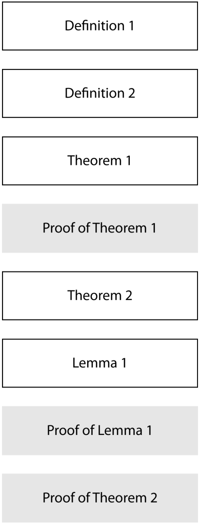
Example of a MathLang Path
Kamareddine, Maarek, Retel and Wells 2007a
Kamareddine, Wells and Zengler 2008



The automatic generation of a proof skeleton Kamareddine, Wells and Zengler 2008

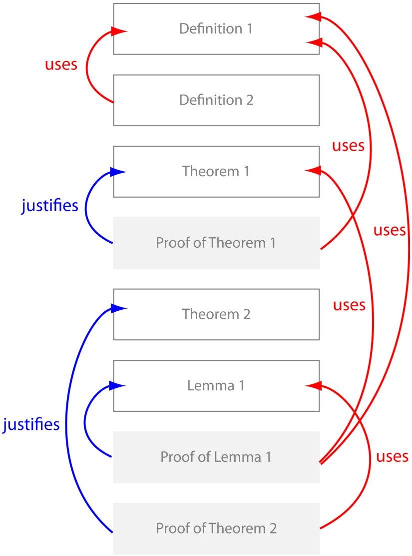
Different provers have

- different syntax
- different requirements to the structure of the text
  e.g.
  - no nested theorems/lemmas
  - only backward references
  - ...
- Aim: Skeleton should be as close as possible to the mathematician's text but with re-arrangements when necessary

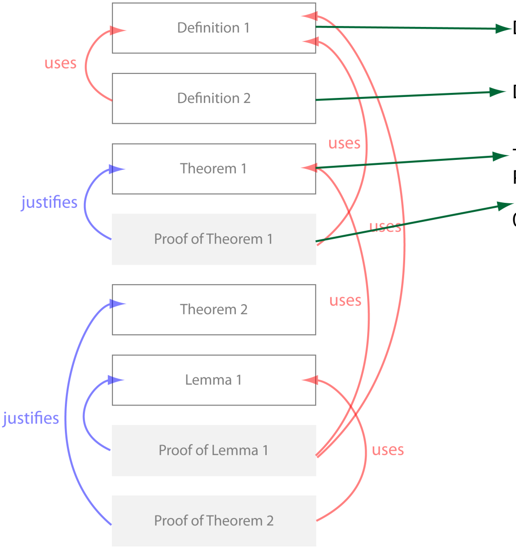


Example of nested theorems/lemmas

The DG for the example

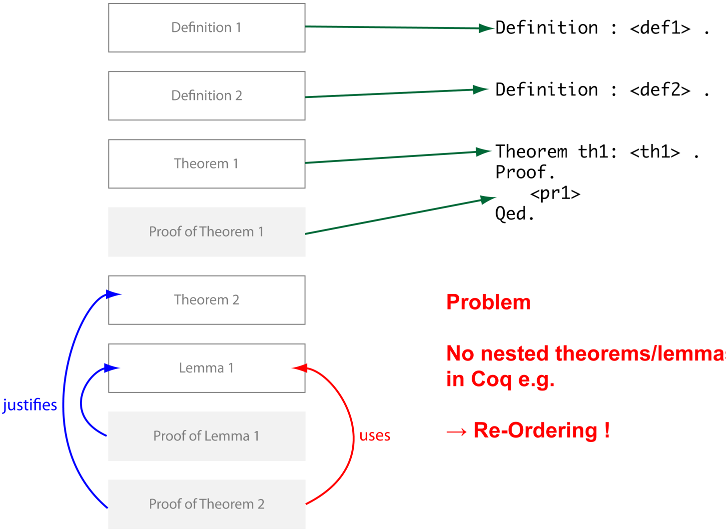


Straight-forward translation of the first part




```
Definition : <def1> .

Definition : <def2> .

Theorem th1: <th1> .
Proof.
    <pr1>
Qed.
```

Problem: nested theorems

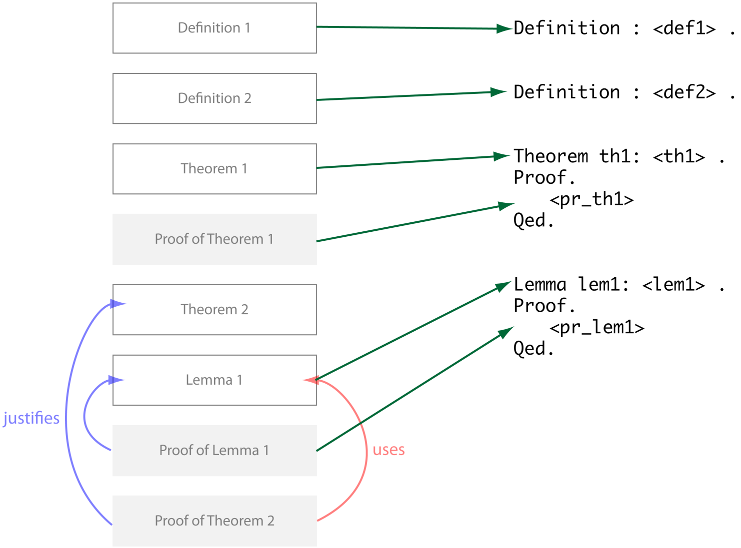


Problem

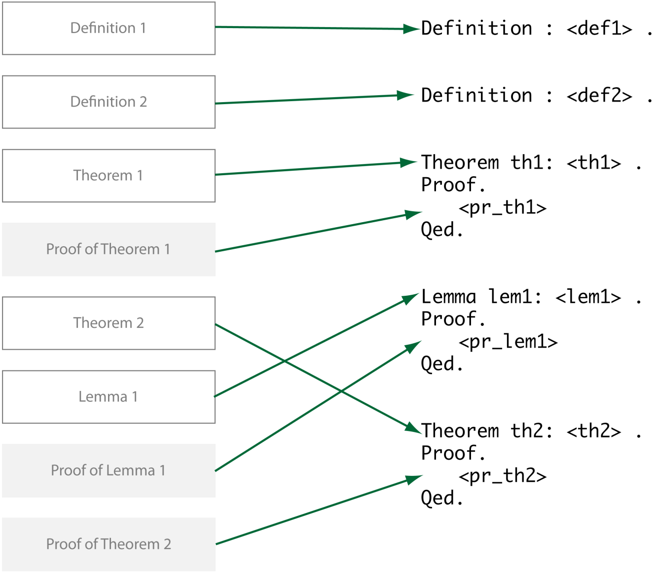
No nested theorems/lemmas in Coq e.g.

→ Re-Ordering !

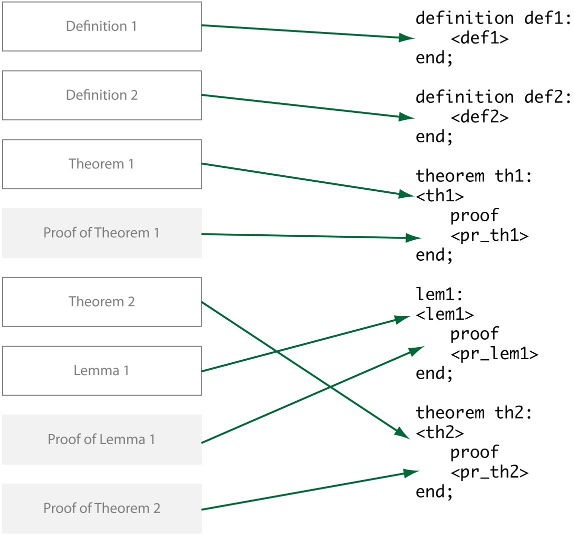
Solution: Re-ordering



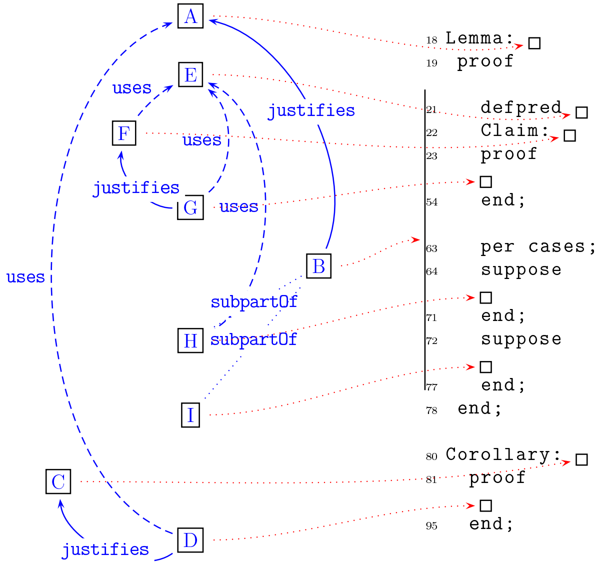
Finishing the skeleton



Skeleton for Mizar



DRa annotation into Mizar skeleton for Barendregt's example (Kamareddine, Maarek, Retel and Wells 2007a)



The remaining very rough path into Mizar Kamareddine, Maarek, Retel, Wells 2007a Kamareddine, Wells, Zengler 2008

- We have not built the remaining aspects all the way into Mizar, but we have a rough path.
- Recall that GoTO gives a Mizar skeleton of the text.
- Next, the CGa encoding of the text is used to build relevant parts of the Mizar FPS (Wiedijk 2003) of the text (e.g., the CGa **preamble** could be used to find counterparts in Mizar MML and to build parts of the *Environment* in Mizar).
- At this stage, a Mizar expert would be able to complete the Mizar FPS version of the text.
- Now, the Mizar experts can complete the formalisation by filling all the gaps in the reasoning (i.e., filling the holes in sentences labelled with the error `*4` by the Mizar system.)

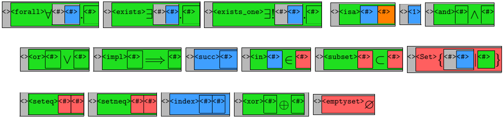
The Mizar FPS version of Barendregt's example

```
Lemma:  for m,n being Nat holds
               m^2 = 2*n^2 implies m = 0 & n = 0
 proof
   let m,n being Nat;
   defpred P[Nat] means
           ex n being Nat st $1^2 = 2*n^2 & $1 > 0;
   Claim: for m being Nat holds
          P[m] implies ex m' being Nat st m' < m & P[m']
   proof
     let m being Nat;
     assume P[m];
     then consider n being Nat such that
     m^2 = 2*n^2 & m > 0;
     m^2 is even ;
::>            *4
     m is even;
::>          *4
     consider k being Nat such that m = 2*k;
::>                                      *4
     2*n^2 = m^2
::>            *4
          .= 4*k^2;
::>              *4
     then n^2 = 2*k^2;
     m > 0 implies m^2 > 0 & n^2 > 0 & n > 0;
::>                                       *4,4,4
     then P[n];
::>          *4,4
     m^2 = n^2 + n^2;
::>              *4
     n^2 + n^2 >  n^2;
::>               *4
     then m^2 > n^2;
::>              *4
     then m > n;
::>           *4
     take m' = n;
     thus thesis;
::>            *4,4
   end;
   A2: for k being Nat holds not P[k]
   proof
     not ex q being Seq_of_Nat
         st q is infinite decreasing by Claim;
::>                                         *4
     hence thesis;
::>            *4
   end;
   assume A0: m^2 = 2*n^2;
   per cases by A0;
   suppose B1: m <> 0;
     then m > 0;
::>          *4
     then P[m] by B1;
::>            *4
     then contradiction by A2;
     hence thesis;
   end;
   suppose S1: m = 0;
     then n = 0;
::>          *4
     thus thesis by S1;
::>              *4
   end;
 end;

Corollary:  sqrt 2 is irrational
  proof
    assume sqrt 2 is rational;
    then ex p,q being Integer st
    q <> 0 & sqrt 2 = p/q;
::>                *4
    then consider m,n being Integer such that
    A0: sqrt 2 = m/n & m = abs m & n = abs n & n <> 0;
::>                                                   *4
    m^2 = 2*n^2;
::>         *4
    n = 0 by Lemma;
::>       *4
    hence contradiction;
::>                *4
  end;

::> 4: This inference is not accepted
```

A full formalisation in Coq via MathLang: chapter 1 of Landau's "Grundlagen der Analysis"

Chapter 1

Natural Numbers



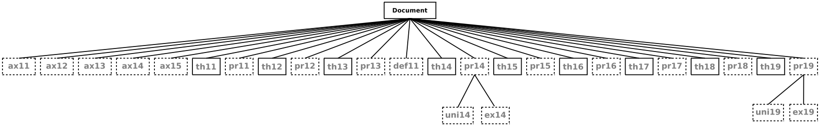
1.1 Axioms

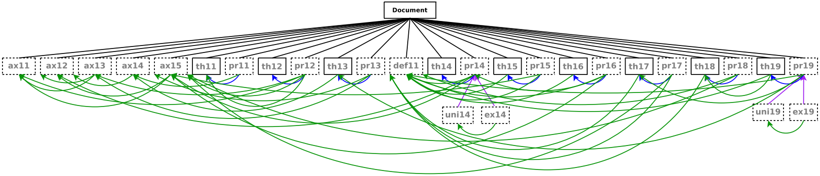
We assume the following to be given:

<><N>A set (i.e. totality) of objects called <><natural_numbers>natural numbers, possessing the properties - called axioms- to be listed below.

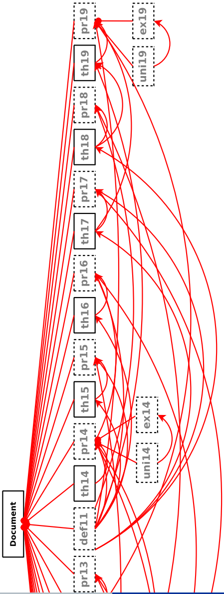
Before formulating the axioms we make some remarks about the symbols $=$ and $\neq$ which be used.

The DRa tree and the DG of sections 1 and 2 of chapter 1 of Landau's book



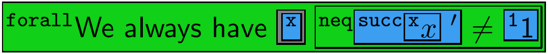


The GoTO of sections 1 and 2 of chapter 1 of Landau's book



Extending proof skeletons with CGa hints

- Translations into Coq for large parts of the chapter can be automated. E.g., the annotation of Axiom 3 ("ax13") is:



By just viewing the interpretations of the annotations we get:

forall x (neq (succ(x), 1)) (a)

The automatically generated Coq proof skeleton for this axiom is:

Axiom ax13 : <ax13> . (b)

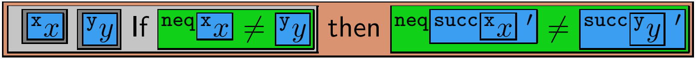
Now, we simply replace the <ax13> placeholder of (b) with the literal translation of the interpretations in (a) to get the valid Coq axiom:

Axiom ax13 : forall x:nats, neq (succ x) I .

Similarly for the theorems of chapter 1 of Landau's book, the work needed to get the full formalisation is straightforward: E.g. Theorem 1 is written by Landau as:

If $x \neq y$ then $x' \neq y'$

Its annotation in MathLang CGa is:



The CGa annotation of the context can also be seen as the premise of an implication. So the upper statement can be translated to:

decl(x), decl(y) : neq x y → neq (succ x) (succ y)

And when we compare this line with its Coq translation we see again, it is just a literal transcription of the interpretation parts of CGa and therefore could be easily performed by an algorithm.

Theorem th11 (x y:nats) : neq x y → neq (succ x) (succ y) .

From the 36 theorems of the chapter 28 could be translated literally into their corresponding Coq theorems.

Some points to consider

- We do not at all assume/prefer one type/logical theory instead of another.
- The formalisation of a language of mathematics should separate the questions:
  - *which type/logical theory is necessary for which part of mathematics*
  - *which language should mathematics be written in*.
- Mathematicians don't usually know or work with type/logical theories.
- Mathematicians usually *do* mathematics (manipulations, calculations, etc), but are not interested in general in reasoning *about* mathematics.
- The steps used for computerising books of mathematics written in English, as we are doing, can also be followed for books written in Arabic, French, German, or any other natural language.

Some points to consider, continued

- MathLang aims to support non-fully-formalized mathematics practiced by the ordinary mathematician as well as work toward full formalization.
- MathLang aims to handle mathematics as expressed in natural language as well as symbolic formulas.
- MathLang aims to do some amount of type checking even for non-fully-formalized mathematics. This corresponds roughly to grammatical conditions.
- MathLang aims for a formal representation of CML texts that closely corresponds to the CML conceived by the ordinary mathematician.
- MathLang aims to support automated processing of mathematical knowledge.

Some points to consider, continued

- MathLang aims to be independent of any foundation of mathematics.
- MathLang allows anyone to be involved, whether a mathematician, a computer engineer, a computer scientist, a linguist, a logician, etc.
- MathLang allows more accurate translation between different languages whithin the mathematical dictionary.

N.G. de Bruijn 1987.
The mathematical vernacular, a language for mathematics with typed sets.
In *Workshop on Programming Logic*, 1987.

G. Cantor 1895.
Beiträge zur Begründung der transfiniten Mengenlehre (Erster Artikel).
Mathematische Annalen, 46:481–512, 1895.

G. Cantor 1897.
Beiträge zur Begründung der transfiniten Mengenlehre (Zweiter Artikel).
Mathematische Annalen, 49:207–246, 1897.

A.-L. Cauchy 1897.
Cours d'Analyse de l'Ecole Royale Polytechnique.
Debure, Paris, 1821.
Also as *Œuvres Complètes* (2), volume III, Gauthier-Villars, Paris, 1897.

R. Dedekind 1872.
Stetigkeit und irrationale Zahlen.
Vieweg & Sohn, Braunschweig, 1872.

G. Frege 1879.
Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens.
Nebert, Halle, 1879.

G. Frege 1892.
Grundgesetze der Arithmetik, begriffschriftlich abgeleitet, volume I.
Pohle, Jena, 1892.

Reprinted 1962 (Olms, Hildesheim).

G. Frege 1903.
Grundgesetze der Arithmetik, begriffschriftlich abgeleitet, volume II.
Pohle, Jena, 1903.
Reprinted 1962 (Olms, Hildesheim).

Heath 1956.
The 13 Books of Euclid's Elements.
Dover, 1956.

Fairouz Kamareddine and J. B. Wells.
PROMATH PResenting, PROving, and PROgramming MATHematical books (Case for Support). 9 pages.
A research proposal to UK funding body, submitted December 2001, acknowledgement G583082.

Fairouz Kamareddine and J. B. Wells.
MATHLANG: A new language for mathematics, logic, and proof checking (Case for Support). 9 pages.
A research proposal to UK funding body, submitted September 2002, acknowledgement 2002102111273723614163.

Kamareddine and Nederpelt 2004.
A refinement of de Bruijn's formal language of mathematics.
J. Logic Lang. Inform., 13(3):287–340, 2004.

Kamareddine, Maarek, and Wells 2005.
Toward an object-oriented structure for mathematical text.

In Michael Kohlhase, editor, *Mathematical Knowledge Management, 4th Int'l Conf., Proceedings*, volume 3863 of *Lecture Notes in Artificial Intelligence*, pages 217–233. Springer, 2006.

Kamareddine, Maarek, Retel and Wells 2007b.
Narrative Structure of Mathematical Texts
Sixth International Conference on MATHEMATICAL KNOWLEDGE MANAGEMENT, 27-30 June, 2007, Linz, Austria. Lecture Notes in AI 4573, Pages 296-312, M. Kauers, Manfred Kerber, Robert Miner and Wolfgang Windsteiger (Eds), ISBN 978-3-540-73083-5. Springer-Verlag, June 2007.

Kamareddine, Maarek, Retel and Wells 2007a.
Gradual computerisation/formalisation of mathematical texts into Mizar.
In *From Insight to Proof, Festschrift in honour of Andrzej Trybulec*, Roman Matuszewski and Anna Zalewska (eds), Studies in Logic, Grammar and Rhetoric, Volume 10(23), Pages 95-120, University of Bialystok, Polish Association for Logic and Philosophy of Science, 2007.

Kamareddine, Lamar, Maarek and Wells 2007. Fairouz Kamareddine, Robert Lamar, Manuel Maarek, and J. B. Wells.
Restoring natural language as a computerised mathematics input method. Sixth International Conference on MATHEMATICAL KNOWLEDGE MANAGEMENT, 27-30 June, 2007, Linz, Austria. Lecture Notes in AI 4573, Pages 280-295, M. Kauers, Manfred Kerber, Robert Miner and Wolfgang Windsteiger (Eds), ISBN 978-3-540-73083-5. Springer-Verlag, June 2007.

Landau 1930.
Grundlagen der Analysis.

Chelsea, 1930.

Landau 1951.
Foundations of Analysis.
Chelsea, 1951.
Translation of [Lan30] by F. Steinhardt.

Moller 2003
General topology. Author's notes.
http://www.math.ku.dk/~moller/e03/3gt/notes/gtnotes.pdf

G. Peano 1989.
Arithmetices principia, nova methodo exposita.
Bocca, Turin, 1889.

Gierz, Hofmann, Keimel, Lawson, Mislove, and Scott 1980.
A Compendium of Continuous Lattices.
Springer-Verlag, 1980.

Wiedijk 2003.
Formal proof sketches.
In *Proceedings of TYPES'03*, volume 3085 of *LNCS*, pages 378–393.
Springer-Verlag, December 2004.