

Computerisation versus Formalisation for Mathematical Texts

FAIROUZ KAMAREDDINE

MathLang: Project started in 2000 by
Fairouz Kamareddine and J.B. Wells.

Ph.D. students in the MathLang team:

M. Maarek, K. Retel, R. Lamar, C. Zengler

Numerous undergraduate and MSc students since 2000

Heriot-Watt University, Edinburgh, UK

<http://www.macs.hw.ac.uk/~fairouz/>

4 April 2008

Some background

- Two questions influence this work:
 - 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, 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 . \square

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 , $\text{\TeX}_{\text{MACS}}$, 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

- + Systems like \LaTeX and $\text{\TeX}_{\text{MACS}}$ provide good defaults for visual appearance, while allowing fine control when needed.
- + \LaTeX and $\text{\TeX}_{\text{MACS}}$ 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.

L^AT_EX 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 x , and \mathfrak{M} be the set of all x for which the assertion holds.

```
\begin{enumerate}
```

item We have $x+1=y'$, 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, Coq, Mizar, 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	X
public documents	X
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 *freedom of movement* between mathematics, logic and computation.
- At every stage, we must have *the choice* of the level of formality 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.)

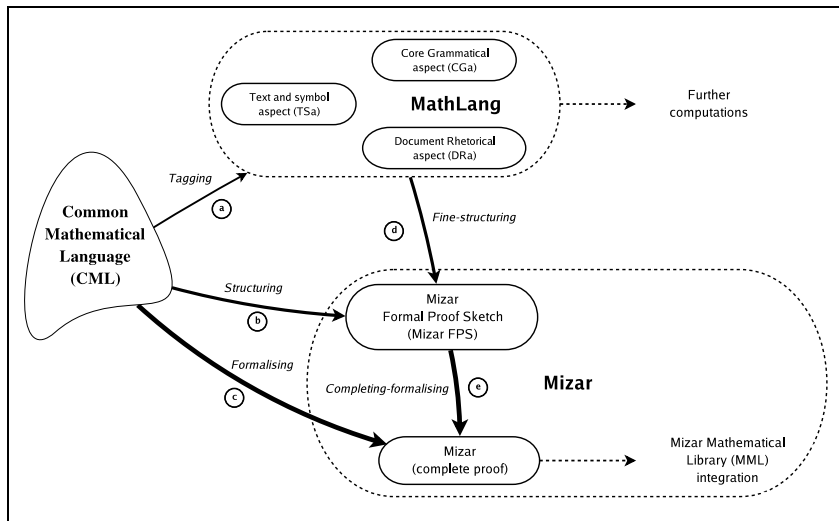
MathLang

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 (into Mizar)

(Kamareddine, Maarek, Retel and Wells 2007a)



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 ...

- 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 et al 1980) as a start.

CGa's grammatical categories

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 $\boxed{\text{an element } 0}$ in \boxed{R} such that $\boxed{a + 0} = a$.

- 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 \mathbb{R} such that $a + (-a) = 0$ for all a in \mathbb{R} .

Another example

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

Another example

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

Another example

There is an element $-a$ in \mathbb{R} such that

$$a + (-a) = 0 \text{ for all } a \text{ in } \mathbb{R}.$$

Another example

There is an element $-a$ in \mathbb{R} such that

$$a + (-a) = 0 \text{ for all } a \text{ in } \mathbb{R}.$$

Another example

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

Another example

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

Another example

There is an element $-a$ in \mathbb{R} such that

$a + (-a) = 0$ for all a in \mathbb{R} .

Another example

There is an element $-a$ in \mathbb{R} such that

$$a + (-a) = 0 \text{ for all } a \text{ in } \mathbb{R}.$$

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.

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

- 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 \mathcal{M} be a set,

y and x are natural numbers,

if x belongs to \mathcal{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 \mathcal{M} be a set,

y and x are natural numbers,

if x belongs to \mathcal{M}

then $x + y$

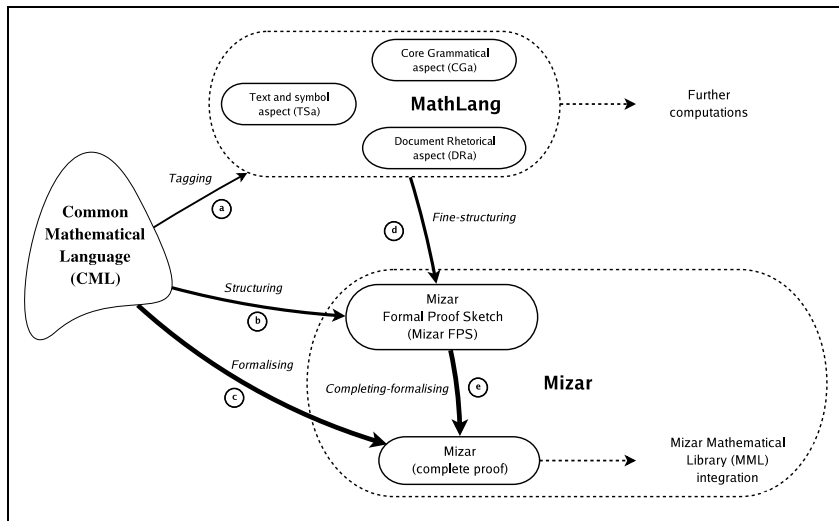
\leftarrow error

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 (into Mizar)

(Kamareddine, Maarek, Retel and Wells 2007a)



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

There is \exists an element 0 in \mathbb{R} such that $\text{eq}(\text{plus } a + 0, a)$

$\{ 0 : \mathbb{R}; \text{eq}(\text{plus}(a, 0), a); \}$

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 assigning intended logical meanings to parts of the text.

Interpretations

There is $\boxed{0}$ an element $\boxed{0}$ in $\boxed{\mathbb{R}R}$ such that $\boxed{\text{eq plus } a a + \boxed{0}} = \boxed{a a}$.

$\{ \ 0 : \mathbb{R}; \ \text{eq} (\ \text{plus} (\ a, \ 0 \), \ a \); \ };$

There is $\boxed{0}$ an element $\boxed{0}$ in $\boxed{\mathbb{R}R}$ such that $\boxed{\text{eq plus } a a + \boxed{0}} = \boxed{a a}$.

There is $\boxed{0}$ an element $\boxed{0}$ in $\boxed{\mathbb{R}R}$ such that $\boxed{\text{eq plus } a a + \boxed{0}}$ equals $\boxed{a a}$.

$\boxed{0} \in \boxed{\mathbb{R}R}, \ \boxed{\text{eq plus } a a + \boxed{0}} = \boxed{a a}$.

Rewrite rules enable natural language representation

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

Diagram illustrating shared sub-expressions in a natural language representation of the equation $0 + a0 = a0 = a(0 + 0) = a0 + a0$. The original equation is shown in blue boxes. Below it, the same equation is shown with green boxes around the terms and pink boxes around the shared sub-expressions $a0$ and $a(0 + 0)$. The word "eq" is written in green next to each equality sign.

How do you do this?

$$0+a0 =$$

$$a0 =$$

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

How do you do this?

$$0+a0 =$$

$$a0 =$$

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

How do you do this?

$$0+a0 =$$

$$a0 =$$

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

How do you do this?

$$\boxed{0+a0} = \langle \text{share} \rangle \boxed{a0} \equiv \langle \text{share} \rangle \boxed{a(0+0)} = \boxed{a0+a0}$$

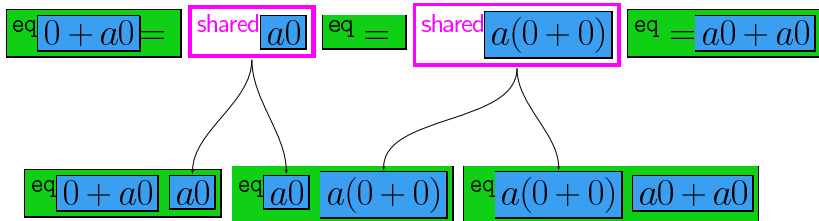
How do you do this?

$$0+a0 = \langle \text{share} \rangle a0 = \langle \text{share} \rangle a(0+0) = a0+a0$$

$$0+a0 = a0$$

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

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



How complete is TSa?

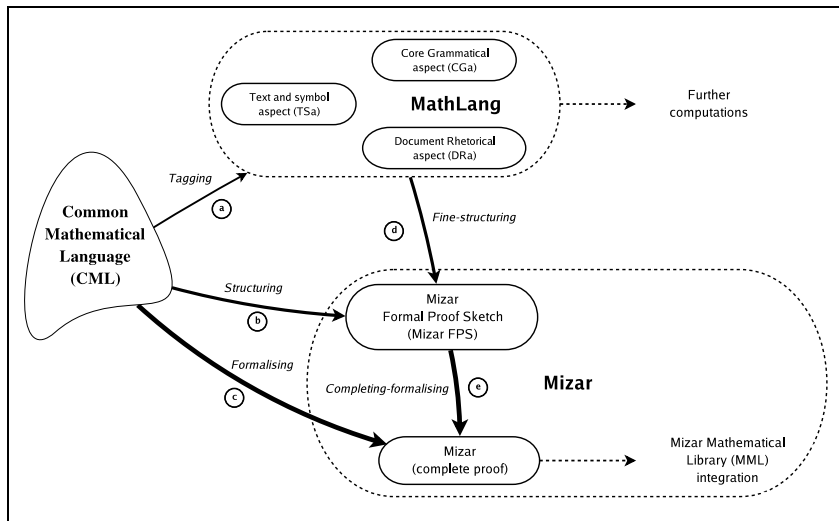
- TSa provides useful interface facilities but it is still under development.
- So far, only simple rewrite (sourcing) rules are used and they are not comprehensive. E.g., unable to cope with things like

$$\overbrace{x = \dots = x}^{n \text{ times}}$$

- The TSa theory and metatheory need development.

Example of a MathLang Path (into Mizar)

(Kamareddine, Maarek, Retel and Wells 2007a)



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
<i>Instances of the StructuralRhetoricalRole class:</i> preamble, part, chapter, section, paragraph, etc.
<i>Instances of the MathematicalRhetoricalRole class:</i> lemma, corollary, theorem, conjecture, definition, axiom, claim, proposition, assertion, proof, exercise, example, problem, solution, etc.
Relation
<i>Types of relations:</i> 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 $m > 0$, it 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: $m^2 = 2n^2 \implies \boxed{\text{A}} n = n = 0$

Proof.

Define on \mathbb{N} the predicate:

$$P(m) \iff \exists n. m^2 = 2n^2 \ \& \ m > 0. \quad \boxed{\text{E}}$$

Claim. $P(m) \implies \boxed{\text{F}} < m. P(m')$.

Indeed suppose $m^2 = 2n^2$ and $m > 0$. It follows that m^2 is even, but then m must be even, as odds squared $\boxed{\text{G}}$ odds. So $m = 2k$ and we have $2n^2 = m^2 = 4k^2 \implies n^2 = 2k^2$. Since $m > 0$, it 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 $m' = n$. $\boxed{\text{B}}$

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 $\boxed{\text{H}}(n)$. Contradiction.

Therefore $m = 0$. But then also $n = \boxed{\text{I}}$ \square

Corollary 1. $\sqrt{\boxed{\text{C}}} \notin \mathbb{Q}$

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 $m = |p|, n = |q| \neq 0$. $\boxed{\text{D}}$ It follows that $m^2 = 2n^2$. But then $n = 0$ by the lemma. Contradiction shows that $\sqrt{2} \notin \mathbb{Q}$. \square

(*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: $m^2 = 2n^2 \iff \boxed{A} \iff n = 0$

Proof.

Define on \mathbb{N} the predicate:

$$P(m) \text{ uses } \boxed{E} \iff \exists n. m^2 = 2n^2 \ \& \ m > 0.$$

justifies

Claim. $P(m) \implies \boxed{E} < m. P(m')$ uses

Indeed suppose $m^2 = 2n^2$ and $m > 0$. It follows that m^2 is even, but then m must be even, justifies \boxed{G} odd. So $m = 2k$ and we have $2n^2 = m^2 = 4k^2 \implies n^2 = 2k^2$ Since $m > 0$, it 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 $m' = n$. \boxed{B}

By the claim $\forall m \in \mathbb{N}. \neg P(m)$, since there are subpartOf descending sequences of natural numbers. subpartOf

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

with $m \neq 0$. Then $m > 0$ and hence $\boxed{H}(n)$. Contradiction.

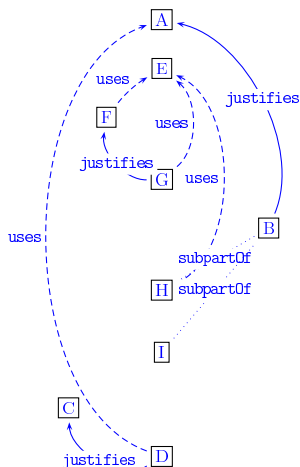
Therefore $m = 0$. But then also $n = \boxed{I}$ \square

Corollary 1. $\sqrt{2} \notin \mathbb{Q}$ \boxed{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 $m = p, n = q$ justifies $\boxed{D} \neq 0$. It follows that $m^2 = 2n^2$. But then $n = 0$ by the lemma. Contradiction shows that $\sqrt{2} \notin \mathbb{Q}$. \square

The automatically generated dependency Graph

Dependency Graph (DG)

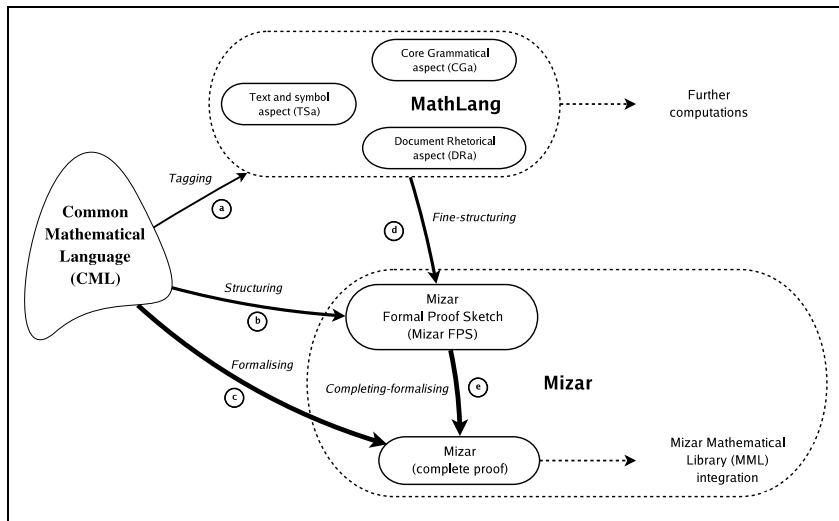


How complete is DRa?

- 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 (into Mizar)

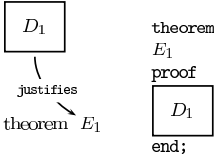
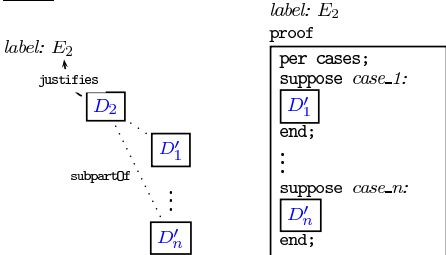
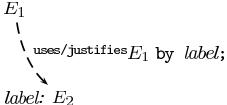
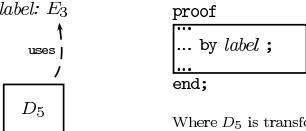
(Kamareddine, Maarek, Retel and Wells 2007a)



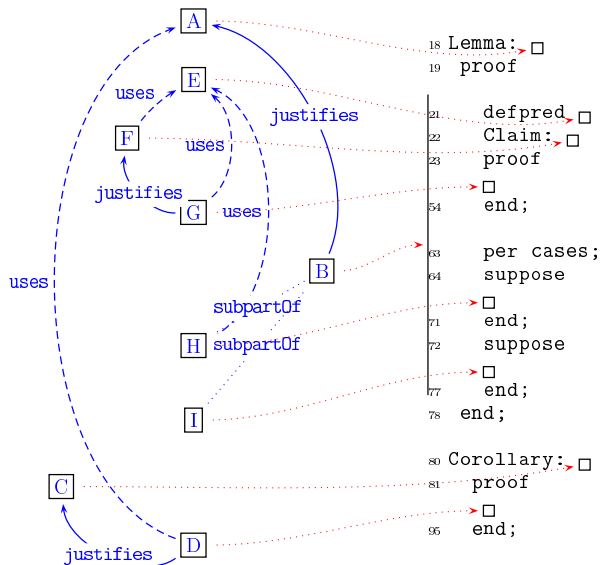
The remaining very rough path into Mizar (Kamareddine, Maarek, Retel and Wells 2007a).

- We have not built the remaining aspects all the way into Mizar, but we have a rough path.
- First, DRa annotations of a text and its automatically generated dependency graph are used to create via a number of transformation hints, a Mizar FPS *Text-Propser* 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.)

Transformation hints from DRa annotations to Mizar skeletons (Kamareddine, Maarek, Retel and Wells 2007a)

<p><u>Hint 1</u></p>  <p>theorem E_1 proof D_1 end;</p>	<p><u>Hint 2</u></p>  <p>label: E_2 proof per cases; suppose <i>case_1</i>: D'_1 end; ... suppose <i>case_n</i>: D'_n end; end;</p> <p>Where D_2 is transformed into box between proof and end;</p>
<p><u>Hint 3</u></p>  <p>E_1 uses/justifies E_1 by <i>label</i>; label: E_2</p>	<p><u>Hint 4</u></p>  <p>label: E_3 proof ... by <i>label</i> ; end;</p> <p>Where D_5 is transformed into box between proof and end;</p>

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



MathLang preamble as subset of Mizar environment for Barendregt's example (Kamareddine, Maarek, Retel and Wells 2007a)

not and \Rightarrow contradiction

\forall \exists

0 2 4

\mathbb{N} \mathbb{Q} \mathbb{Z}

$=$ \neq $>$ $<$

\in \notin is

2 $\sqrt{\quad}$ $*$ $+$ $/$ $||$

$-$ $\{ \}$ sequence of

number even infinite descending

```
7 vocabularies INT_1, SQUARE_1, MATRIX_2, IRRAT_1,
8   RAT_1, ARYTM_3, ABSVALUE, SEQM_3, FINSET_1;
9 notations INT_1, NAT_1, SQUARE_1, XXREAL_0,
10  ABIAN, RAT_1, IRRAT_1, XCMLPX_0, INT_2, SEQM_3,
11  FINSET_1, REAL_1, PEPIN;
12 constructors INT_1, NAT_1, SQUARE_1, XXREAL_0,
13  ABIAN, RAT_1, IRRAT_1, XCMLPX_0, INT_2, SEQM_3,
14  FINSET_1, PEPIN;
15 requirements SUBSET, NUMERALS, ARITHM, BOOLE, REAL;
16 registrations XREAL_0, REAL_1, NAT_1, INT_1;
```

The Mizar FPS version of Barendregt's example

```
20 Lemma: for m,n being Nat holds
21       m^2 = 2*n^2 implies m = 0 & n = 0
22 proof
23   let m,n being Nat;
24   defpred P[Nat] means
25     ex n being Nat st $1^2 = 2*n^2 & $1 > 0;
26 Claim: for m being Nat holds
27   P[m] implies ex m' being Nat st m' < m & P[m']
28 proof
29   let m being Nat;
30   assume P[m];
31   then consider n being Nat such that
32     m^2 = 2*n^2 & m > 0;
33     m^2 is even ;
34 ::> *4
35   m is even;
36 ::> *4
37   consider k being Nat such that m = 2*k;
38 ::> *4
39     2*n^2 = m^2
40 ::> *4
41       . = 4*k^2;
42 ::> *4
43     then n^2 = 2*k^2;
44     m > 0 implies m^2 > 0 & n^2 > 0 & n > 0;
45 ::> *4,4,4
46     then P[n];
47 ::> *4,4
48     m^2 = n^2 + n^2;
49 ::> *4
50     n^2 + n^2 > n^2;
51 ::> *4
52     then m^2 > n^2;
53 ::> *4
54     then m > n;
55 ::> *4
56     take m' = n;
57     thus thesis;
58 ::> *4,4
59   end;
```

```
67   A2: for k being Nat holds not P[k]
68   proof
69     not ex q being Seq_of_Nat
70       st q is infinite decreasing by Claim;
71 ::> *4
72   hence thesis;
73 ::> *4
74   end;
75   assume A0: m^2 = 2*n^2;
76   per cases by A0;
77   suppose B1: m <> 0;
78     then m > 0;
79 ::> *4
80     then P[m] by B1;
81 ::> *4
82     then contradiction by A2;
83     hence thesis;
84   end;
85   suppose S1: m = 0;
86     then n = 0;
87 ::> *4
88     thus thesis by S1;
89 ::> *4
90   end;
91 end;
```

```
93 Corollary: sqrt 2 is irrational
94 proof
95   assume sqrt 2 is rational;
96   then ex p,q being Integer st
97     q <> 0 & sqrt 2 = p/q;
98 ::> *4
99   then consider m,n being Integer such that
100     A0: sqrt 2 = m/n & m = abs m & n = abs n & n <> 0;
101 ::> *4
102     m^2 = 2*n^2;
103 ::> *4
104     n = 0 by Lemma;
105 ::> *4
106     hence contradiction;
107 ::> *4
108   end;
```

```
109
110 ::> 4: This inference is not accepted
```

Much more work needed on the MathLang path into Mizar

- The MathLang path after the DRa annotation and checking stage, has been given informally and in a very rough manner.
- We would like to develop full blown aspects that clearly and precisely take the text into full Mizar.
- Aspects must be well developed (theory, implementation, automation and consistency between aspects).
- Immediate attention is needed to make the transformation hints into a more precise calculus and an automated algorithm be given. Full automation may not be possible, but a good balance of interaction between the user and the computer needs to be created.
- The passage from CGa annotated parts of text into Mizar text parts (e.g., Preamble into Environment) needs to be formalised and a balance between user and computer created.
- Open question: what aspects are needed to pass into full Mizar (including what aspects can transform a Mizar FPS text into full Mizar text).

Other MathLang paths

- A current PhD project (with student Lamar), is to work on the MathLang path into Isabelle.
- After the CGa, and DRa annotations of the text (using TSa as far as possible), and after obtaining a number of correctness checks on the text (grammatical, relational, DG, etc), it would be interesting to check at which stage the path into Mizar differs from the path into Isabelle and how much can we keep in common.
- Although we are currently working on MathLang paths into Isabelle and Mizar, we are open to other provers. We will be assessing both direct paths and paths through translations.

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 within 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, begriffsschriftlich abgeleitet, volume I.

Pohle, Jena, 1892.

Reprinted 1962 (Olms, Hildesheim).



G. Frege 1903.

Grundgesetze der Arithmetik, begriffsschriftlich abgeleitet, volume II.

Pohle, Jena, 1903.

Reprinted 1962 (Olms, Hildesheim).



Heath 1956.

The 13 Books of Euclid's Elements.

Dover, 1956.



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.



G. Peano 1889.

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.