MathLang: A language for Mathematics

Fairouz Kamareddine

`www.macs.hw.ac.uk/~fairouz/talks/talks2004/kings04.pdf`[*]

18 February 2004

[*]Parts of this talk are based on joint work with Nederpelt [4] and Maarek and Wells [5]

The 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.

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* [Lan51].

Theorem 6. [Commutative Law of Addition]

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

Proof Fix y, and $\mathfrak{M}$ 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. □

LaTeX code

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 problem with formal logic

- *Frege, Begriffsschrift [1]*: *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 precision*

- In 1879, he wrote the *Begriffsschrift*, whose *first purpose is to provide us with the most reliable test of the validity of a chain of inferences*.

- He wrote the *Grundlagen* and *Grundgesetze der Arithmetik* where mathematics is seen as a branch of logic and arithmetic is described in *Begriffsschrift*.

- In 1902, Russell wrote a letter to Frege [1] informing him of a *paradox* (see [3]).

- To avoid the paradox, Russell used *type theory* in the famous *Principia Mathematica* [7] where mathematics was founded on logic.

- Advances were also made in *set theory* [8], *category theory* [6], etc., each being advocated as a better foundation for mathematics.

- But, none of the logical languages of the 20th century satisfies the criteria expected of a language of mathematics.
  - A logical language does not have *mathematico-linguistic* categories, is *not universal* to all mathematicians, and is *not a satisfactory 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 writes in logic their *understanding* 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.

The problem with fully checked proofs (on computer)

- 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.).

- A CML-text is structured differently from a computer-checked text proving the same facts. *Making the latter involves extensive knowledge and many choices:*
  - First, the needed choices include:
    - 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 system*: a type theory (dependent?), a set theory (ZF? FM?), a category theory? etc.
    - The choice of the *proof checker*: Automath, Isabelle, Coq, PVS, Mizar...
  - Any informal reasoning in a CML-text will cause headaches as it is hard to turn a big step into a (series of) syntactic proof expressions.
  - Then the CML-text is *reformulated* in a fully *complete* syntactic formalism where every detail is spelled out. Very long expressions replace a clear CML-text. The new text is useless to ordinary mathematicians.
- So, *automation is user-unfriendly* for the mathematician/computer scientist.
- It is the hope that the alternative to CML may help in dividing the jump from informal mathematics to a fully formal one into smaller more informed steps.

Coq

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.
```

Where do we start? de Bruijn's Mathematical Vernacular MV

- *De Bruijn's Automath not just [...] as a technical system for verification of mathematical texts, it was rather a life style with its attitudes towards understanding, developing and teaching mathematics.... The way mathematical material is to be presented to the system should correspond to the usual way we write mathematics. The only things to be added should be details that are usually omitted in standard mathematics.*

- MV is 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 to remain *closer to* a given informal mathematical content.

- We want a *formal* language MathLang which •has the advantages of CML but not its disadvantages and •respects CML content.

- The above items mean that MV fails in this aim.

What is the aim for MathLang?

Can we formalise a CML 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 CML text (and hence the content of the original CML text is respected).

2. The formalised text can be fully manipulated and searched.

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 as simple a process to the mathematician as LaTeX is.

Starting point for MathLang: MV and WTT

- MV is the driving force behind MathLang. But MV fails on goal 1.
- Weak Type Theory, WTT [4], started from MV, but attempted to avoid its problems.
- WTT was intended as a *a 2nd language* for mathematicians which *satisfies* many criteria:

a1. WTT is *formal, suitable for computerization.*

a2. WTT helps mathematicians precisely identify the structure where they work.

a3. WTT makes an expert/teacher/student *aware of the complexity of a mathematical notion.*

a4. WTT encourages thinking about the interdependencies of notions (e.g., in which part of the chapter the definition holds), *so WTT texts are better structured than* CML *texts.*

a5. WTT respects all linguistic categories in the special ways they are used by mathematicians, e.g., *nouns*, *adjectives*, etc.

a6. WTT does *not restrict* the mathematician to set/type/category theory.

a7. Unlike set/type theory, WTT has basic notions needed for *text* such as *definition*, *theorem*, *step in a proof*, *section*, etc.

a8. The ambiguities in the CML-texts disappear in the translation to WTT. For example, the anaphoric obscurities in CML are resolved in WTT by the strict context management.

a9. Although the CML text and its initial translation into WTT are incomplete, WTT has *additional* levels *supporting more rigor.*

One can define *further translations into more logically complete versions.*

Improvements of MathLang over WTT

- Although WTT succeeds in many ways and is a considerable improvement on MV, it still fails on goal 1. A WTT text is not close to its CML original.

- MathLang starts from WTT, extends its syntax and adds natural language as a top level.

- A *MathLang text remains* close *to its* CML *original and hence yields* reliable *formalizations.*
  The CML-text is covered exactly in its formal version in the MathLang-text.
  One can easily check this.

- We are using MathLang to translate two CML-books [Lan51, Hea56]

MathLang

draft documents	✓
public documents	✓
computations and proofs	✓

- MathLang describes the grammatical and reasoning structure of mathematical texts
- A *weak type system* checks MathLang documents at a grammatical level
- MathLang eventually should support *all encoding uses*

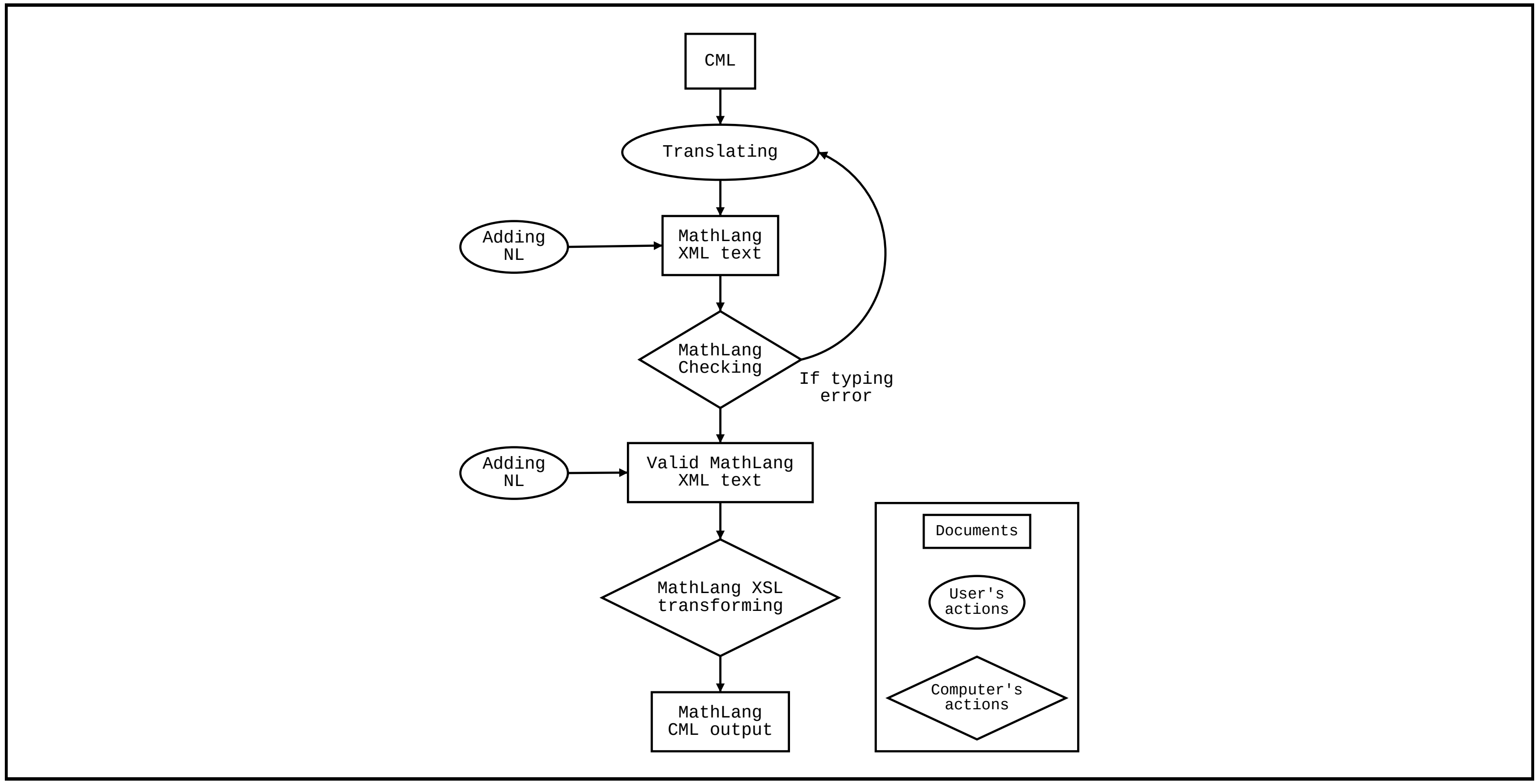

Figure 1: Translation

MathLang's Grammatical categories

They extend those of WTT with blocks and flags.

`T`	terms
`S`	sets
`N`	nouns
`A`	adjectives
`P`	statements
`D`	definitions
`Z`	declarations
Γ	contexts with flags
`L`	lines
`K`	blocks
`B`	books

The Grammatical Categories of MathLang

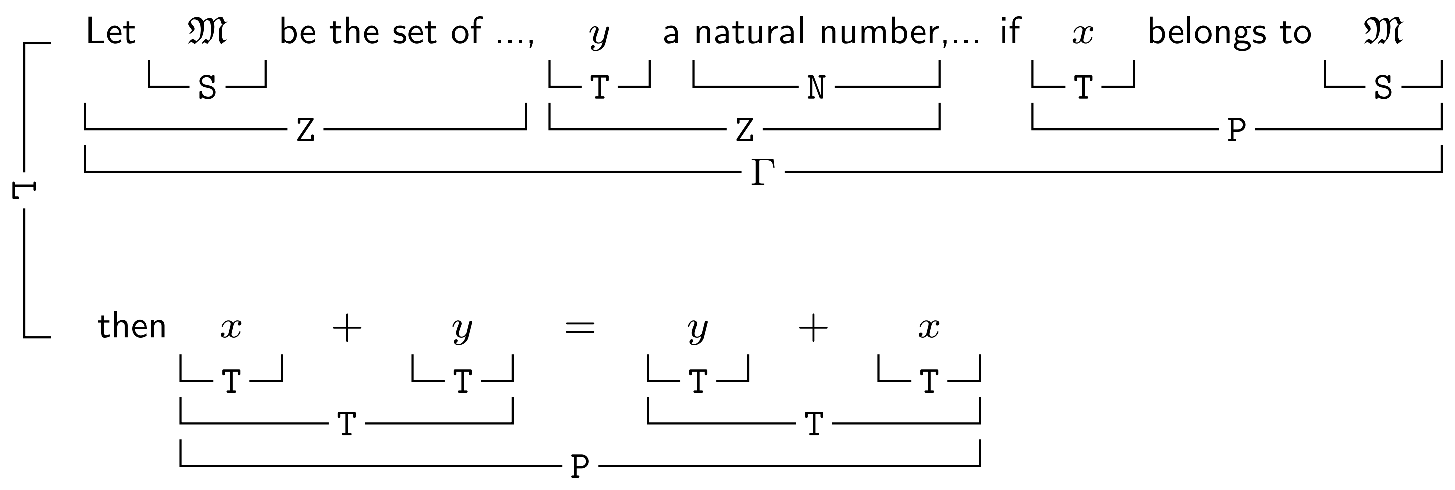
Let $\mathfrak{M}$ be the set of ..., y a natural number,... if x belongs to $\mathfrak{M}$
S
T
N
T
S
Z
Z
P
Γ
L
then $x + y = y + x$
T
T
T
T
T
T
P

Figure 2: A mathematical line and its grammatical categories

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:

$$\begin{array}{lll} B;\Gamma\ \vdash\ t :: T, & B;\Gamma\ \vdash\ s :: S, & B;\Gamma\ \vdash\ n :: N, \\ B;\Gamma\ \vdash\ a :: A, & B;\Gamma\ \vdash\ p :: P, & B;\Gamma\ \vdash\ d :: D \end{array}$$

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

A preface for a book B could look like:

constant name	weak type	constant name	weak type
$\mathbb{R}$	S	$\cup$	$S \times S \rightarrow S$
$\surd$	$T \rightarrow T$	$\geq$	$T \times T \rightarrow P$
$+$	$T \times T \rightarrow T$	$\wedge$	$P \times P \rightarrow P$

- $\mathbb{R}$ has no parameters and is a set.
- $\surd$ is a constant with one parameter, a term, delivering a term.
- $\geq$ is a constant with two parameters, terms, delivering a statement.
- $\texttt{prefcons}(B) = \{\mathbb{R}, \surd, +, \cup, \geq, \wedge\}$.

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

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

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

$$\frac{}{\vdash\ \emptyset :: \mathbf{B}} \quad (emp{-}book)$$

$$\frac{B;\Gamma \ \vdash\ p :: P}{\vdash\ B \circ \Gamma \triangleright p :: \mathbf{B}} \qquad \frac{B;\Gamma \ \vdash\ d :: D}{\vdash\ B \circ \Gamma \triangleright d :: \mathbf{B}} \quad (book{-}ext)$$

Example

CML: the square root of the third power of a natural number

MathLang: $\mathtt{Abst}_{n:\mathbb{N}}(\sqrt{n^3})$

The preface is:

	constant name	weak type
(i)	^{3}	$T \to T$
(ii)	$\sqrt{}$	$T \to T$
(iii)	$\mathbb{N}$	S
(iv)	$\mathtt{Abst}$	$T \to N$

The categories are:

subexp	category	subexp	category	subexp	category
n	T	n	T	$\mathtt{Abst}_{n:\mathbb{N}}(\sqrt{n^3})$	N
n^3	T	$\mathbb{N}$	S		
$\sqrt{n^3}$	T	$n : \mathbb{N}$	$\mathcal{Z}$		

We need to derive $B;\Gamma \vdash \mathtt{Abst}_{n:\mathbb{N}}(\sqrt{n^3}) :: N$ for some B and Γ.

But it is clear that $B = \Gamma = \emptyset$.

(1)		$\vdash$	$\emptyset :: \mathbf{B}$	$(emp{-}book)$
(2)	$\emptyset$	$\vdash$	$\emptyset :: \mathbb{T}$	$(emp{-}cont, 1)$
(3)	$\emptyset; \emptyset$	$\vdash$	$\mathbb{N} :: S$	$(ext{-}cons, 1, 2, iii)$
(4)	$\emptyset$	$\vdash$	$n : \mathbb{N} :: \mathbb{T}$	$(term{-}decl, 1, 2, 3, *)$
(5)	$\emptyset; n : \mathbb{N}$	$\vdash$	$n :: T$	$(var, 1, 4, *)$
(6)	$\emptyset; n : \mathbb{N}$	$\vdash$	$n^3 :: T$	$(ext{-}cons, 1, 4, i, 5)$
(7)	$\emptyset; n : \mathbb{N}$	$\vdash$	$\sqrt{n^3} :: T$	$(ext{-}cons, 1, 4, ii, 6)$
(8)	$\emptyset; \emptyset$	$\vdash$	$\mathtt{Abst}_{n:\mathbb{N}}(\sqrt{n^3}) :: N$	$(bind, 1, 4, iv, 7)$

Figure 3: Derivation that $\mathtt{Abst}_{n:\mathbb{N}}(\sqrt{n^3})$ is a noun

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).

Comparaison with other work

- No theorem provers provide an independent language for describing mathematical content in such a manner that the *goals 1..4* are sufficiently accounted for.

- *Existing mathematical vernaculars are not ready for immediate use*, and if accessible for a mathematical user, then with great difficulty.

- *Galina* usable as a *language of commands* for Coq. Not meant as *a first step in formalization* as MathLang is.

- The built-in version of the *mathematical vernacular* of *ΩMEGA* is meant to give the user on request a mathematics-like computer *view* of an already checked proof. It has the same drawbacks as Galina.

- The basic languages of *Mizar* and Isar are close to the reliability criterion and have proven to be suited for expressing large corpora of mathematical content.

Their syntax is, however, rather complicated and requires much of an ordinary user to become acquainted with it.

- In the *Theorema project* computer algebra systems, the provers are designed to imitate the proof style humans employ in their proving attempts. The proofs can be produced in human-readable style. However, this is done by *post-processing* a formal proof in natural language.

- The typed functional programming language *GF* defines languages such as fragments of natural languages, programming languages and formal calculi. GF is based on Martin-Löf's type theory.

  We do not at all assume/prefer one type theory instead of another.

- The formalisation of a language of mathematics should separate the questions:
  - *which type theory is necessary for which part of mathematics*
  - *which language should mathematics be written in*.
- Mathematicians don't usually know or work with type theories.
- Mathematicians usually *do* mathematics (manipulations, calculations, etc), but are not interested in general in reasoning *about* mathematics.

MathLang example

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 $x + y = y + x$

MathLang 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 $x + y$ $\Leftarrow$ **error**

MathLang example

blocks flags references

Theorem 6. [Commutative Law of Addition]

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

Proof Fix y, and $\mathfrak{M}$ 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.

MathLang skeleton

$x : \mathbb{N},\ y : \mathbb{N}\ \rhd \mathsf{Th6}(x, y) := x + y = y + x$ | | (97)

Proof Theorem 6 | | {2.5.4}

Proof Theorem 6 part I | | | {2.5.4.1}

$\boxed{y : \mathbb{N}}$ | | | |

| $\boxed{\mathfrak{m} : \mathsf{SET}}$ | | | |

| | $\boxed{\forall_{x:\mathfrak{m}} \mathsf{Th6}(x, y)}$ | | | |

| | | $(\mathsf{Def}\ +(38))\ \rhd y + 1 = y'$ | | | | (98)

| | | $\{2.5.1\}\ \rhd 1 + y = y'$ | | | | (99)

| | | $(98), (99)\ \rhd 1 + y = y + 1$ | | | | (100)

| | | $(100)\ \rhd \mathsf{Th6}(1, y)$ | | | | (101)

| | | $(101)\ \rhd 1 : \mathfrak{m}$ | | | | (102)

| | | *Proof Theorem 6 part II* | | | {2.5.4.2}

| | | $\boxed{x : \mathfrak{m}}$ | | | |

| | | | $\mathsf{Th6}(x, y)\ \rhd x + y = y + x$ | | | | (103)

| | | | $(103)\ \rhd (x + y)' = (y + x)'$ | | | | (104)

| | | | $(\mathsf{Def}\ +(39))\ \rhd (y + x)' = y + x'$ | | | | (105)

| | | | $(104), (105)\ \rhd (x + y)' = y + x'$ | | | | (106)

| | | | $\{2.5.2\}\ \rhd x' + y = (x + y)'$ | | | | (107)

| | | | $(107), (\mathsf{Def}\ +(39))\ \rhd x' + y = y + x'$ | | | | (108)

| | | | $(108)\ \rhd \mathsf{Th6}(x', y)$ | | | | (109)

| | | | $(109)\ \rhd x' : \mathfrak{m}$ | | | | (110)

| | | $\mathsf{Ax5}(\mathfrak{m}, (102), (110))\ \rhd \mathbb{N} \subset \mathfrak{m}$ | | | (111)

$(111)\ \rhd \forall_{x:\mathbb{N}} \forall_{y:\mathbb{N}} \mathsf{Th6}(x, y)$ | | | (112)

References

[Hea56] Heath. *The 13 Books of Euclid's Elements*. Dover, 1956.

[1] Heijenoort, J. v. (ed.): 1967, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*. Cambridge, Massachusetts: Harvard University Press.

[2] D.T. van Daalen. *The Language Theory of Automath*. PhD thesis, Eindhoven University of Technology, 1980.

[3] Kamareddine, F., L. Laan, and R. Nederpelt: 2002, 'Types in logic and mathematics before 1940'. *Bulletin of Symbolic Logic* **8**(2), 185–245.

[4] Kamareddine, F., and R. Nederpelt: 2004, A refinement of de Bruijn's formal language of mathematics. Journal of *Logic, Language and Information*. Kluwer Academic Publishers.

[5] Kamareddine, F., Maarek, M., and Wells, J.B.: 2004, MathLang: An experience driven language of mathematics, Electronic Notes in Theoretical Computer Science 93C, pages 123-145. Elsevier.

[Lan30] Edmund Landau. *Grundlagen der Analysis*. Chelsea, 1930.

[Lan51] Edmund Landau. *Foundations of Analysis*. Chelsea, 1951. Translation of [Lan30] by F. Steinhardt.

[6] MacLane, S.: 1972, *Categories for the Working Mathematician*. Springer.

[7] Whitehead, A. and B. Russell: 1910^1, 1927^2, *Principia Mathematica*, Vol. I, II, III. Cambridge University Press.

[8] Zermelo, E.: 1908, 'Untersuchungen über die Grundlagen der Mengenlehre'. *Math. Annalen* **65**, 261–281.