Toward an Object-Oriented Structure for Mathematical Text

Fairouz Kamareddine, Manuel Maarek and Joe Wells

ULTRA Group - Heriot-Watt University

November 14th, 2005 Theorema-ULTRA-Omega'05 Workshop Saarland University, Germany

イロト イポト イヨト イヨト

Computerising mathematical texts

CML

The Common Mathematical Language used by mathematicians in their everyday writings is known as *meticulous*, in comparison with other natural languages, *flexible*, in its way to accommodate many branches of mathematics, *coherent* by providing contextual justifications of statements.

Is CML reflected in current approaches of computerising Mathematics?

イロト イポト イヨト イヨト

Two examples From Euclid to Bourbaki

Definition 20. Of trilateral figures, an equilateral triangle is that which has its three sides equal, an isosceles triangle that which has two of its sides alone equal, and a scalene triangle that which has its three sides unequal.

Euclid [The 13 Books of Euclid's Elements, Book I]

・ロン ・回 と ・ ヨン ・ ヨン

Definition 1. A set with an associative law of composition, possessing an identity element and under which every elements is invertible, is called a group. [...] A group G is called finite if the underlying set of G is finite [...] A group [with operators] G is called commutative (or Abelian) if its group law is commutative.

N. Bourbaki [Elements of Mathematics - Algebra, volume II, Chapter I, §4]

Two examples From Euclid to Bourbaki

Definition 20. Of trilateral figures, an equilateral triangle is that which has its three sides equal, an isosceles triangle that which has two of its sides alone equal, and a scalene triangle that which has its three sides unequal.

Euclid [The 13 Books of Euclid's Elements, Book I]

・ロン ・回 と ・ ヨン ・ ヨン

Definition 1. A set with an associative law of composition, possessing an identity element and under which every elements is invertible, is called a group. [...] A group G is called finite if the underlying set of G is finite [...] A group [with operators] G is called commutative (or Abelian) if its group law is commutative.

N. Bourbaki [Elements of Mathematics - Algebra, volume II, Chapter I, §4]

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

Mathematical word processing ${}_{{\mathbb E}^{\!X}}$



```
\begin{definition}
    Of trilateral figures, an equilateral triangle is that which has its
    three sides equal, an isosceles triangle that which has two of its
    sides alone equal, and a scalene triangle that which has its three
    sides unequal.
    \end{definition}
```

```
\begin{definition}
```

```
A set with an associative law of composition, possessing an identity
element and under which every elements is invertible, is called a
group. [...] A group $G$ is called finite if the underlying set of
$G$ is finite [...] A group [with operators] $G$ is called
commutative (or Abelian) if its group law is commutative.
\end{definition}
```

Visual representation of CML

Difficult semantic recognition

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

Semantic markup languages MathML, OpenMath, OMDoc

${\sf OpenMath}/{\sf OMDoc}$

- Flexible
- Difficult semantic recognition due to the mixture of structural and symbolic XML and chunks of natural language.
- Extensible with embedded formal content

Full formalisation

Theorem provers

Our goal differs from full formalisation.

We want to provide a control over presentation and phrasing of the semantic structure. Most mathematical texts are unlikely to be formalized, but might well benefit from computerisation.

Procedural style – such as Coq, Isabelle

- Fully formalised
- Requires expertise
- Formalisation that may not reflect the CML text

Declarative style - such as Mizar

Fully formalised

Mathematical word processing

Computerising the mathematical vernacular

Semantic markup languages Full formalisation

- Requires expertise and the Mizar Mathematical Library
- Syntax mimics natural language
- Formal Proof Sketch (a lighter version of Mizar)

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

Computerising the mathematical vernacular N.G. de Bruijn's MV – WTT – MathLang-WTT – MathLang

N.G. de Bruijn's Mathematical Vernacular A language with substantives, adjectives and flags

The Weak Type Theory A type system for MV with weak types (TERM, NOUN, ADJ, SET, STAT, LINE and BOOK)

MathLang-WTT A practical evaluation of MV and WTT

- Extends WTT with FLAGS and BLOCKS
- Automates type checking

 $\top \cup \Omega$

- Has been used to translate existing CML texts
- Proposes various output-views faithful to CML

MathLang's approach to computerise mathematical texts is to:

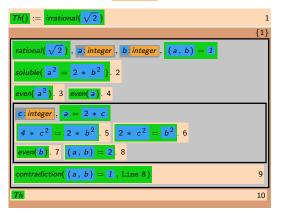
- Capture, in a first ground, the grammatical structure of the text
- Enhance this first ground language with a choice of features (semantical, logical, ...)

(ロ) (同) (E) (E) (E)

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

Computerising the mathematical vernacular MathLang-WTT – output-view (Example from F. Wiedijk's comparison)

T Terms S Sets N Nouns A Adjectives P Statements Z Declarations F Contexts L Lines F Flags K Blocks B Books

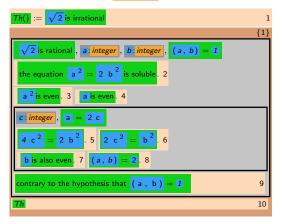


- Symbolic view
- CML view of symbols
- CML view of the document

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

Computerising the mathematical vernacular MathLang-WTT – output-view (Example from F. Wiedijk's comparison)

T Terms S Sets N Nouns A Adjectives P Statements Z Declarations F Contexts L Lines F Flags K Blocks B Books

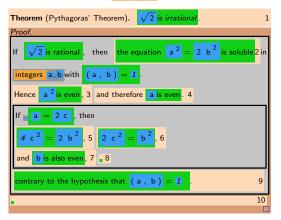


- Symbolic view
- CML view of symbols
- CML view of the document

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

Computerising the mathematical vernacular MathLang-WTT – output-view (Example from F. Wiedijk's comparison)

T Terms S Sets N Nouns A Adjectives P Statements Z Declarations F Contexts L Lines F Flags K Blocks B Books



- Symbolic view
- CML view of symbols
- CML view of the document

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

イロン イヨン イヨン イヨン

MathLang-WTT

 $\top \cup \Omega$

Encodings of Euclid's and Bourbaki's examples?

How to faithfully encode a triangle and its sides, a group and its law in MathLang-WTT?

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

イロン イヨン イヨン イヨン

MathLang-WTT Encodings of Euclid's and Bourbaki's examples?

How to faithfully encode a triangle and its sides, a group and its law in MathLang-WTT?

triangle and side, group and law as constants of type NOUN.

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

・ロン ・回と ・ヨン・

MathLang-WTT Encodings of Euclid's and Bourbaki's examples?

How to faithfully encode a triangle and its sides, a group and its law in MathLang-WTT?

triangle and side, group and law as constants of type NOUN.

How to encode the intrinsic relation between a triangle and its lines and between a group and its law?

Mathematical word processing Semantic markup languages Full formalisation Computerising the mathematical vernacular

MathLang-WTT Encodings of Euclid's and Bourbaki's examples?

How to faithfully encode a triangle and its sides, a group and its law in MathLang-WTT?

triangle and side, group and law as constants of type NOUN.

How to encode the intrinsic relation between a triangle and its lines and between a group and its law?

- By parametrising triangle and group with sides and law
 → Constraining & not flexible
- By using a statement "has".

```
has(triangle,line1); has(triangle,line2); has(triangle,line3)
has(group,law)
```

\rightarrow Verbose & not reliable

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives

Back to N.G. de Bruijn's informal definitions.

MV's substantives (MathLang-WTT's nouns)

MV's adjectives (MathLang-WTT's adjectives)

MV's names (MathLang-WTT's terms)

イロト イポト イヨト イヨト

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives

- Back to N.G. de Bruijn's informal definitions.
- Analogy with Object-oriented programming.

MV's substantives (MathLang-WTT's nouns) Classes

MV's adjectives (MathLang-WTT's adjectives) Mixins (functions from classes to classes)

MV's names (MathLang-WTT's terms) **Objects**

 $\top \cup \Omega$

イロト イポト イヨト イヨト

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives

- Back to N.G. de Bruijn's informal definitions.
- Analogy with Object-oriented programming.
- New design of MathLang with object-oriented aspects.
- MV's substantives (MathLang-WTT's nouns)

Classes

Nouns as classes

MV's adjectives (MathLang-WTT's adjectives) Mixins (functions from classes to classes) Adjectives as mixins

MV's names (MathLang-WTT's terms)

Objects

Terms as objects

・ロン ・回と ・ヨン・

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives Euclid's example

Definition 20. Of trilateral figures, an equilateral triangle is that which has its three sides equal, an isosceles triangle that which has two of its sides alone equal, and a scalene triangle that which has its three sides unequal. Euclid [The 13 Books of Euclid's Elements, Book I]

Figure and triangle defined as nouns. Trilateral and equilateral defined as adjectives.

イロト イヨト イヨト イヨト

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives Euclid's example

Definition 20. Of trilateral figures, an equilateral triangle is that which has its three sides equal, an isosceles triangle that which has two of its sides alone equal, and a scalene triangle that which has its three sides unequal. Euclid [The 13 Books of Euclid's Elements, Book I]

Figure and triangle defined as nouns. Trilateral and equilateral defined as adjectives.

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives Bourbaki's example

 $\top \cup \Omega$

Definition 1. A set with an associative law of composition, possessing an identity element and under which every elements is invertible, is called a group. [...] A group G is called finite if the underlying set of G is finite [...] A group [with operators] G is called commutative (or Abelian) if its group law is commutative. N. Bourbaki [Elements of Mathematics - Algebra, volume II, Chapter I, §4]

Group defined as a noun. Finite and Abelian defined as adjectives.

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives Bourbaki's example

 $\top \cup \Omega$

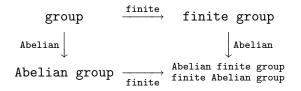
Definition 1. A set with an associative law of composition, possessing an identity element and under which every elements is invertible, is called a group. [...] A group G is called finite if the underlying set of G is finite [...] A group [with operators] G is called commutative (or Abelian) if its group law is commutative. N. Bourbaki [Elements of Mathematics - Algebra, volume II, Chapter I, §4]

Group defined as a noun. Finite and Abelian defined as adjectives.

Object-oriented approach Examples Multi adjective refinements

Abstraction with nouns and adjectives

Multi adjective refinements



- Combine the adjectives *finite* and *Abelian* to obtain either Abelian finite group or finite Abelian group.
- In MathLang both expressions share the same type. Their meaning may differ as the statements introduced by the adjectives may overlap.
- It is possible to define an isosceles equilateral scalene triangle.
- But not a Abelian triangle (with these current definitions).

÷.

Syntax Type system Example

. . . .

・ロン ・回 と ・ ヨン ・ ヨン

æ

Syntax Sets, category expressions and identifiers

ıdent, ı	=	denumerably infinite set of identifiers	
label, l	=	denumerably infinite set of labels	
cvar, v	=	denumerably infinite set of category variables	
category, c	::=	term(exp) set(exp) noun(exp) adj(exp, exp)	
		<pre>stat dec(category) cvar</pre>	
cident, ci		ident exp.cident	
ciuent, ci	::=	ident exp.cident	

• •

. .

.. . .

Syntax Type system Example

Syntax Steps

 $\top \cup \Omega$

 $\begin{array}{rcccccc} step, \ s & ::= & phrase & & Basic unit \\ & & | & label \ label \ step & Labelling \\ & | & step \ \triangleright \ step & Local \ scoping \\ & | & \{\overrightarrow{step}\} & & Block \end{array}$

(an arrow on top of a meta-variable represents a sequence of 0 or more meta-variables)

(ロ) (同) (E) (E) (E)

Syntax Type system Example

Syntax Phrases and expressions

 $\top \cup \Omega$

phrase, p	::=	exp	
		$cident(\overrightarrow{ident}) := exp$	Definition
	Í	$ident(\overrightarrow{exp}) := exp$	Definition by matching case
		ident \ll cident	Sub-noun and adjective statement
exp, e	::=	cident(exp)	Instance
		ident(category) : exp	Elementhood declaration
		ident(category) : category	Declaration
	1 I	Noun {step}	Noun
		Adj(exp) {step}	Adjective
		exp exp	Refinement
		self super	Self and super
		ref label	Referencing

▲□→ ▲圖→ ▲厘→ ▲厘→

æ

Syntax Type system Example

Type system Rules for steps

$$\frac{\vdash s_{1} : Step}{s_{1} \vdash s_{2} : Step} \quad \{s_{1}; s_{2}\} \vdash \{\vec{s}\} : Step}{s_{1} \vdash \{s_{2}; \vec{s}\} : Step} \text{ STEP-COMPOSITION}$$

$$\frac{\vdash s : Step}{s \vdash s' : Step} \quad \{s; s'\} \vdash s'' : Step}{s \vdash s' \vdash s'' : Step} \text{ LOCAL-SCOPING}$$

$$\frac{\vdash s : Step}{s \vdash p : Step} \quad \text{ATOMIC-STEP}$$

$$\frac{\vdash \{\} : Step}{\vdash \{\} : Step} \text{ EMPTY-STEP}$$

Is Toward an Object-Oriented Structure for Mathematical Text

イロン イヨン イヨン イヨン

æ

Syntax Type system Example

Type system Rules for noun and adjective expressions

$$\begin{array}{c|c} \vdash s \mathrel{\bullet} Step & \{s; self : Term(T)\} \vdash s' \mathrel{\bullet} Step \\ \hline \forall i \in I(s'), \{s; self : Term(T); s'\} \vdash i \mathrel{\bullet} T(i) \\ \hline s \vdash \text{Noun} \{s'\} \mathrel{\bullet} Noun(T) \end{array} \text{NOUN}$$

$$\begin{array}{l} \vdash s : Step \qquad s \vdash e : Noun(T) \\ T \leq T' \qquad \{s; super : Term(T); self : Term(T')\} \vdash s' : Step \\ \forall i \in I(s'), \{s; super : Term(T); self : Term(T'); s'\} \vdash i : T'(i) \\ \hline s \vdash \mathbf{Adj} (e) \{s'\} : Adj(T, T') \end{array}$$
ADJ

$$\begin{array}{c|c} \vdash s \colon Step & s \vdash e_1 \And Adj(T_1, T_1') \\ \hline s \vdash e_2 \And Noun(T_2)/Set(T_2)/Term(T_2) & T_1 \leq T_2 \\ \hline s \vdash e_1e_2 \And Noun(T_1' \uplus T_2)/Set(T_1' \uplus T_2)/Term(T_1' \uplus T_2) \end{array}$$
REFINEMENT

・ロン ・回 と ・ ヨン ・ ヨン

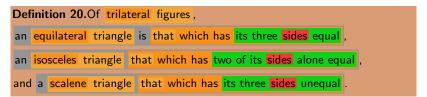
æ

Syntax Type system Example

Type system Example of typing – Euclid's example

 $\top \cup \Omega$

Term Terms Set Sets Noun Nouns Adj Adjectives Stat Statements
Def Definition Step Local scopings ▷ Step Blocks { }



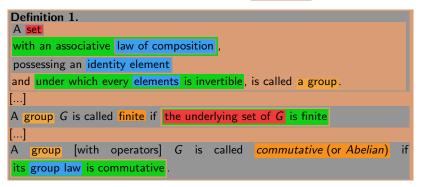
・ロット (四) (日) (日)

Syntax Type system Example

Type system Example of typing – Bourbaki's example

Term Terms Set Sets Noun Nouns Adj Adjectives Stat Statements

Def Definition Step Local scopings ▷ Step Blocks { }



<ロ> (日) (日) (日) (日) (日)

Other works Future works Conclusion

Other works Krzysztof Retel

In his research work, Krzysztof Retel investigates ways to

- Bridge MathLang with existing systems for formalising mathematics.
- Design MathLang features that would extend the semantical knowledge contained in MathLang documents and provide opportunities for verification.

To target these two points he experienced translations of CML documents into Mizar via MathLang. He compared this translation path with a direct translation into Mizar and proposed guidance for such gradual translations.

イロン イヨン イヨン イヨン

Other works Future works Conclusion

Future works MathLang ongoing works

- \blacktriangleright Development of a user interface for MathLang based on the scientific editor $T_{E}X_{\rm MACS}.$
- Design of MathLang extension features.
- Bridging existing systems and languages (Mizar, OpenMath, OMDoc) with features combinations.
- Orienting MathLang development with translation of mathematical documents.
- Refinement of the object-oriented aspect of MathLang with traits.

・ロン ・回 と ・ 回 と ・ 回 と

Other works Future works Conclusion

Conclusion MathLang

- We saw how the experience-driven development of MathLang led us to extend our ground language by turning nouns into classes and adjectives into mixins.
- MathLang provides an expressive encoding for computerising the symbolic and natural language parts of mathematical text.
- Our current work is aimed to demonstrate the utility of decomposing the path towards full formalisation.

イロト イポト イヨト イヨト