

**FROM FORMAL SPECIFICATION TO FULL PROOF:  
A STEPWISE METHOD**

*by*

Lavinia Burski



Submitted for the degree of  
Doctor of Philosophy

DEPARTMENT OF COMPUTER SCIENCE  
SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES  
HERIOT-WATT UNIVERSITY

March 2016

The copyright in this thesis is owned by the author. Any quotation from the report or use of any of the information contained in it must acknowledge this report as the source of the quotation or information.

# Abstract

Write the abstract here.

# Contents

<b>1</b>	<b>Background</b>	<b>1</b>
1.1	MathLang for Mathematics . . . . .	1
1.1.1	The Current MathLang Design . . . . .	2
1.1.2	Core Grammatical aspect . . . . .	3
1.1.3	Text and Symbol aspect . . . . .	4
1.1.4	Document Rhetorical aspect . . . . .	4
1.1.5	Full formalisation paths into Theorem Provers . . . . .	4
1.1.6	Conclusion . . . . .	5
1.2	Formal Notation . . . . .	6
1.2.1	Tools for Z . . . . .	6
	<b>Bibliography</b>	<b>7</b>

# Acronyms

**CGa** Core Grammatical aspect.

**DRa** Document Rhetorical aspect.

**TSa** Text and Symbol aspect.

# Chapter 1

## Background

### 1.1 MathLang for Mathematics

1. *MathLang is a framework.* It is meant to be used for communication as a concrete support for human mind formulation. MathLang is a well structured framework aimed to synthesize the common mathematical language.
2. *MathLang is for mathematics.* It is meant to be open to any branch of mathematics and to any topic that uses mathematics as a base language. MathLang mimics mathematics in its incremental construction of a body of knowledge.
3. *MathLang is for computerisation.* MathLang is meant to be a medium for a human-system, human-human via a digital support, and system-system communication. MathLang is a computer-based framework and therefore offers automation facilities.

Taken from Maarek's thesis [11].

MathLang's original Goals, when the project started in 2000 was to allow a gradual computerisation and formalisation of mathematical texts.

MathLang is not a system for proof verification but a framework to computerise and translate information (such as mathematical text) into a form on which proof checkers can operate.

The MathLang framework provides extra features supporting more rigour to translation of the common mathematical language. One can define further levels of translations into more semantically and logically complete versions. This gradual computerisation method should be more accessible than direct formalisation, because a number of first levels do not require any particular expertise in formalisation.

So far Mathlang has given alternative and complete paths which transform mathematical texts into new computerised and formalised versions. Dividing the formalisation of mathematical texts into a number of different stages was first proposed by N.G. de Bruijn to relate common mathematical language to his Mathematical Vernacular [1] and his proof checking system Automath.

### 1.1.1 The Current MathLang Design

The MathLang Framework instructs the computerisation process to be broken up into a number of levels called **aspects**. Having an aspect prevents the misunderstanding of the process of computerisation of mathematical documents using the MathLang Framework. Each aspect can be worked out independently, simultaneously or sequentially without prior knowledge of another aspect. The current MathLang Framework contains three well-developed aspects, the Core Grammatical aspect (CGa), the Text and Symbol aspect (TSa) and the Document Rhetorical aspect (DRa), and has further aspects such as the Formal Proof Sketch.

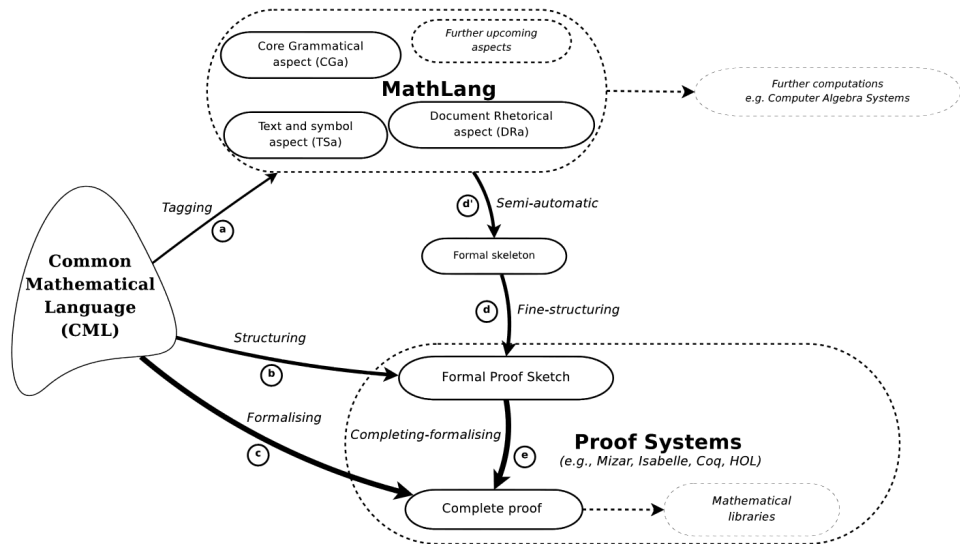


Figure 1.1: The MathLang approach to computerisation/formalisation [4]

Figure 1.1 shows the overall situation of work in the current MathLang Framework. The labelled arrows show the computerisation paths from the common mathematical language to any proof system. The width of the arrow representing each path segment increases according to the expertise required to achieve the path segment. The level of expertise needed to computerise a CML text straight into a complete proof is very high, however the level of expertise is much smaller by using the MathLang framework to help form a formal skeleton and then into a complete proof. The dashed arrows illustrate further computerisation that one can envision.

### 1.1.2 Core Grammatical aspect

The current CGa in MathLang uses a finite set of grammatical *categories* to identify the structure and common concepts used in mathematical texts. The aims of the CGa is to make explicit the grammatical role played by the elements of mathematical texts and to allow the validation of the grammatical and reasoning structure within the CGa encoding in a mathematical text. The CGa checks for grammatical correctness and finds errors like an identifier being used without and prior introduction or the wrong number of arguments being given to a function [12].

### 1.1.3 Text and Symbol aspect

The TSa builds the bridge between a mathematical text and its grammatical interpretation. The TSa is a way of rewriting parts of the text so they have the same meaning. For example some mathematicians may prefer to write " $a=b$  and  $b=c$  and  $c=d$ ", others may prefer " $a=b$ ,  $b=c$ ,  $c=d$ " and some others may prefer " $a=b=c=d$ ". As you can see all these methods of writing have the same meaning however some symbols are different. The TSa annotates each expression in the text with a string of words or symbols which aim to act as the mathematical representation of which this expression is. This allows everything in the text to be uniform.

### 1.1.4 Document Rhetorical aspect

The Document Rhetorical aspects checks that the correctness of the reasoning in the mathematical document is correct and that there are no loops. The DRa mark-up system is simple and more concentrated on the narrative structure of the mathematical documents whereas other previous systems (such as DocBook <sup>1</sup>, Text Encoding Initiative <sup>2</sup>, OMDoc <sup>3</sup>) were more concentrated on the subtleties of the documents. It is used to describe and annotate chunks of texts according to their narrative role played within the document [12]. Using the DRa annotation system we can capture the role that a chunk of text imposes on the rest of the document or on another chunk of text. This leads to generating dependency graphs which play an important role on mathematical knowledge representation. With these graphs, the reader can easily find their own way while reading the original text without the need to understand all of its subtleties. Processing DRa annotations can flag problems such as circular reasoning and poorly-supported theorems.

### 1.1.5 Full formalisation paths into Theorem Provers

The MathLang project starts in 2000 when F.Kamareddine and J.B. Wells started the project within Heriot-Watt University as part of the ULTRA group [2]. It was an

---

<sup>1</sup><http://www.docbook.org>

<sup>2</sup><http://www.tei-c.org/index.xml>

<sup>3</sup><http://www.omdoc.org>



idea for a new mathematical language and framework to keep most of the advantages of Common Mathematical Language (CML) and avoid its disadvantages. This framework would allow a gradual computerisation and formalisation of mathematical texts.

The framework was first set out in 2003 [10] and then saw an established path for conversion of a mathematical text written in CML into the Isabelle proof assistant using rules and MathLang annotations [7]. A few short projects (by 4th year undergraduate dissertations or MSc students) have developed MathLang into the Framework it is today. A prototype of the Core Grammatical aspect and Text and Symbol aspect were defined in the PhD thesis of Manuel Maarek [11] and then a gradual computerisation into the Mizar proof assistant using the three key aspects of MathLang were a great success and published [4].

The design of the CGa is due to Kamareddine, Maarek and Wells [5] and the implementation of the CGa is due to Maarek [11]. The design of the TSa is due to Kamareddine, Maarek, and Wells with contributions by Lamar to the sourcing rules [3], [11], and [8]. The implementation is primarily by Maarek [11] with contributions from Lamar [8]. The design and implementation of DRa were the subject of Retel's thesis [12]. Further additions have since been carried out by Zengler [6].

### **1.1.6 Conclusion**

A lot of work has already been completed on the MathLang Framework. The three aspects, CGa, DRa, and TSa, have been designed and redesigned so that a variety of mathematical texts and symbols could be used. Then the aspects have been implemented for ease of access. A translation from a mathematical text into the Mizar proof checker has been worked through and described in detail in a published paper [4] and a PhD thesis [11]. A partial translation from a CML text into the Isabelle Syntax has also been carefully described in the 2009 paper [9] and also written in a PhD thesis [8]. Some of the future work described was to allow MathLang to be used as a tool to computerise other pieces of information, such as another piece of academic text yet it does not need to be mathematical.

## 1.2 Formal Notation

### 1.2.1 Tools for Z

Write background chapter

write design chapter

write conclusion chapter

# Bibliography

- [1] N. de Bruijn. The mathematical vernacular, a language for mathematics with typed set. *In Workshop on Programming Logic*, 1987.
- [2] F. Kamareddine and J.B.Wells. A research proposal to uk funding body. *Formath*, 2000.
- [3] F. Kamareddine, R. Lamar, M. Maarek, and J. B. Wells. Restoring natural language as a computerised mathematics input method. In M. Kauers, M. Kerber, R. Miner, and W. Windsteiger, editors, *Calculemus/MKM*, volume 4573 of *Lecture Notes in Computer Science*, pages 280–295. Springer, 2007.
- [4] F. Kamareddine, M. Maarek, K. Retel, and J. B. Wells. Gradual computerisation/formalisation of mathematical texts into mizar. In *From Insight to Proof: Festschrift in Honour of Andrzej Trybulec*, pages 81–95. Springer-Verlag, 2007.
- [5] F. Kamareddine, M. Maarek, and J. B. Wells. Toward an object-oriented structure for mathematical text. In M. Kohlhase, editor, *MKM*, volume 3863 of *Lecture Notes in Computer Science*, pages 217–233. Springer, 2005.
- [6] F. Kamareddine, J. B. Wells, and C. Zengler. Computerising mathematical texts in MathLang. Technical report, Heriot-Watt University, 2008.
- [7] R. Lamar. The mathlang formalisation path into isabelle – a second-year report, 2003.
- [8] R. Lamar. *A Partial Translation Path from MathLang to Isabelle*. PhD thesis, Heriot-Watt University, 2011.

- [9] R. Lamar, F. Kamareddine, and J. B. Wells. Mathlang translation to isabelle syntax. In J. Carette, L. Dixon, C. S. Coen, and S. M. Watt, editors, *Calculus/MKM*, volume 5625 of *Lecture Notes in Computer Science*, pages 373–388. Springer, 2009.
- [10] M. Maarek. Mathematical documents faithfully computerised: the grammatical and text & symbol aspects of the mathlang framework, first year report, 2003.
- [11] M. Maarek. *Mathematical documents faithfully computerised: the grammatical and test & symbol aspects of the MathLang Framework*. PhD thesis, Heriot-Watt University, 2007.
- [12] K. Retel. *Gradual Computerisation and Verification of Mathematics: MathLang’s Path into Mizar*. PhD thesis, Heriot-Watt University, 2009.