Magdalena Wolska

A Language Engineering Architecture for Processing Informal Mathematical Discourse

In: Petr Sojka (ed.): Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008. Masaryk University, Brno, 2008. pp. 131--136.

Persistent URL: http://dml.cz/dmlcz/702548

A Language Engineering Architecture for Processing Informal Mathematical Discourse

Magdalena Wolska

Computational Linguistics, Saarland University, Saarbrücken 66040, Germany
E-mail: magda@coli.uni-saarland.de

Abstract. We present a modular architecture for processing informal mathematical language as found in textbooks and mathematical publications. We point at its properties relevant in addressing three aspects of informal mathematical discourse: (i) the interleaved symbolic and natural language, (ii) the linguistic, domain, and notational context, and (iii) the imprecision of the informal language. The objective in the modular approach is to enable parameterisation of the system with respect to the natural language of the text and the mathematical domain of discourse.

Key words: mathematical discourse, language processing

1 Introduction

Informal mathematical discourse in textbooks and mathematical publications is partly written in natural language and partly in a symbolic notation — even within a single utterance. Be it information retrieval or text mining mathematical documents, flexible human-oriented mathematical user interfaces, or automated verification of informal proofs crucially rely on automated analysis of the informal language.

In [8,9,2] we presented methods of, respectively: parsing, lexical analysis, and domain-specific interpretation of informal mathematical proofs, and introduced linguistic resources necessary for processing. In this paper, we present a modular architecture of a system for processing mathematical language based on those resources and emphasise three core aspects of the informal mathematical discourse it addresses: the interleaved symbolic and natural language, the linguistic, domain, and notational context, and the imprecision of the informal language. The objective in the modular approach is to enable parameterisation of the system with respect to the natural language of the text in question, the mathematical domain of the discourse, and the mathematical notation. We first briefly discuss the above-mentioned aspects of the mathematical language based on example utterances, then we present our processing architecture, and finally outline the related work on mathematical discourse and our further work.

Petr Sojka (editor): DML 2008, Towards Digital Mathematics Library, pp. 131–136.
 ISBN 978-80-210-4658-0

2 Language Phenomena

The following statements from naive set theory in English and German illustrate three core re-curring language phenomena; A and B are sets:[1]

(1) If $x \in B$ then $x \notin A$.
Wenn $x \in B$ *dann* $x \notin A$

(2) B contains no $x \in A$.
B *enthaelt kein* $x \in A$

(3) A contains no elements that are also in B.
A *enthaelt keinesfalls Elemente, die auch in* B *sind.*

(4) A and B have no common elements.
A *hat keine Elemente mit* B *gemeinsam.*

Symbolic and Natural Language. The two language "modes" can be smoothly interleaved and the same content can be expressed using a variety of syntactic constructions, more and less verbose. Symbolic expressions can occur as clausal, (1), or noun phrase, (2)–(4), constituents depending on whether they are formulas or terms.[2] (2) shows how tightly the two modes can be interleaved: under the intended reading, the formula $x \in A$ has two constituents ("x", a noun constituent and "$\in$ A", a restrictive relative clause with the meaning "which is an element of A"), the scope of the negation word (here in the function of a a determiner) is over a part of the formula following it ("x" on the left-hand side). Tight interleaving is a cross-linguistic phenomenon, shown here only in English and German. A parsing component of any processing architecture must be capable of systematic analysis of the various degrees of verbalisation and the interactions between the natural and symbolic languages.

Linguistic and Domain Context, and Notation. The linguistic context is determined by the national language in which the text is written (e.g. English, German) with its lexis and grammar. The lexicon includes general and specialised (technical) vocabulary. Certain general words have technical meaning in mathematics (the word "group" for example) and should be part of the terminological dictionary. A number of technical terms may be compounds or multi-word proper names (e.g. names of theorems). Certain multi-word constructs have fixed meaning (e.g. "if and only if," "without loss of generality").

The domain context is specified globally by the mathematical domain of the discourse (e.g. naive set theory) and locally by the given text segment (e.g. a proof or its meaningful part: a case in a case-split, a step in an inductive

[1] The presentation of language phenomena is limited here for space reasons. Further discussion can be found in the mentioned publications. Examples therein are from tutorial dialogues, however, the phenomena are also relevant in processing textbooks.

[2] This is, of course, an over-generalisation: formula *mentions* occur in the function of nouns/noun phrases in formal logic texts, for instance, on proof theory. "$P \supset Q$ is a formula" is a typical example from a paragraph on the syntax of propositional logic.

proof). The global domain context determines the scope of the terminological dictionary, whereas the local domain context determines the interpretation of the introduced entities (e.g. their types; A and B above are sets — they could be sentence parameters in another context) and the scope of reference (e.g. for anaphor resolution).

The notational convention determines the set of symbols used and the way symbolic expressions are to be "read" (i.e. parsed) and interpreted, as well as syntactic constructions and referring expressions; when pre- or post-fix notation is used, constructs such as (2) and referring expressions such as "the left-/right-hand side" (of a formula) are not likely to occur.

Imprecision. While mathematics is generally considered as the precise science par excellence, its language can be remarkably imprecise. This is for the most part due to the natural language part which tends to introduce ambiguity and imprecision. (2) and (3) exemplify just one type of lexical ambiguity introduced by the word "contain" which may be interpreted in the sense of membership or subset (lexical ambiguity at the domain level). Common referring expressions such as "the left side" or "the smaller set" are also imprecise; the former typically refers to the object denoted by the term to the left of some formula's main operator, rather than the area with respect to its geometric center, the latter typically denotes the set of lower cardinality, rather than a set of a smaller physical size. One of the sources of these types of imprecision are the mental models which we employ in thinking about mathematical concepts, e.g. conceptual metaphors. These, in turn, can manifest themselves in the language. Because mental models are an inherent part of mathematical thinking [6], imprecise language is only to be expected in informal mathematical discourse and hence must be addressed in a mathematical discourse processing system. In particular, the example utterances above (1–4) should obtain the same resulting interpretation.

3 The Architecture

In this section we outline the architecture of our discourse processing system which can analyse mathematical utterances including the ones exemplified in Section 2. The system is built on a pipe-line architecture which consists of three larger sub-parts: pre-processing, parsing, and sentence level and discourse level interpretation (see Figure 1) and is parameterised with respect to the natural language of the text, the mathematical domain, and the mathematical notation convention. The input to the system is raw ascii text with mathematical symbols marked by their unicodes or in LaTeX(-like) format. The components marked with downward diagonal lines are those whose resources are dependent on the natural language, upward lines mark dependence on mathematical domain, and crossing lines on both. The processing proceeds as follows:

At the **pre-processing** stage the text is **segmented** into sentences and these into word(-like) tokens using language and domain specific regular expressions

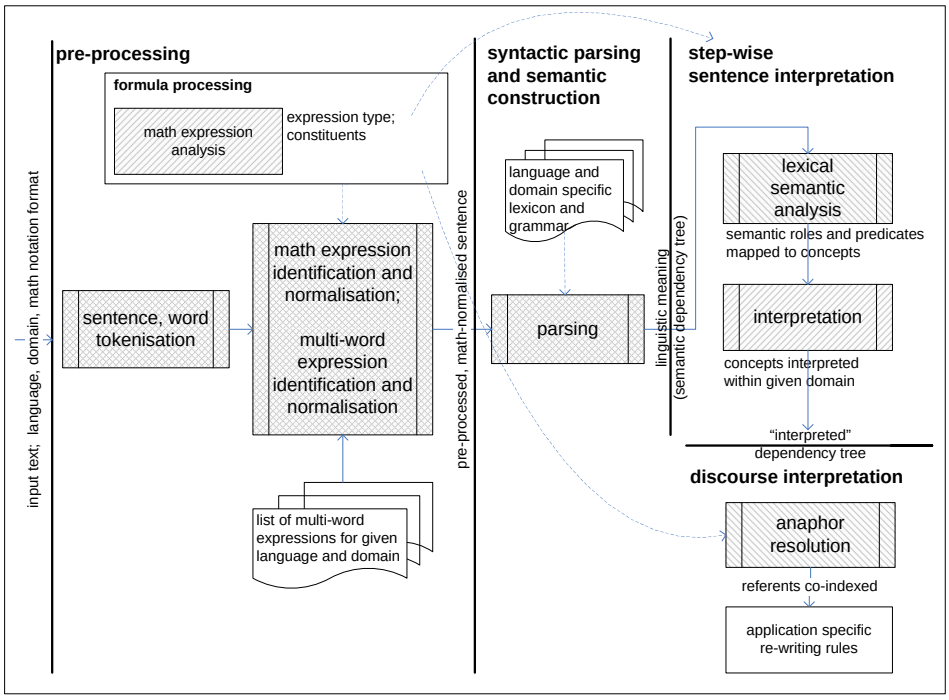

Fig. 1. The processing architecture

(e.g. set of relevant mathematical symbols). **Multi-word expression**s (MWEs) are identified by regular expressions (currently language and domain specific lists of MWEs, including wording variants, are read from a file). Likewise, **mathematical expressions** (MEs) are identified: the ME tagger takes word-tokenised text as input and finds ME substrings using regular expressions. This process is again language and domain specific (for instance, in English, "a" is ambiguous between an indefinite article and a mathematical symbol). Once identified, MEs are parsed. Since parsing MEs itself is not the main focus of our work, we implemented only a simple precedence-based parser which builds ME trees. An external component, based on for example [4], can be integrated into the architecture, so long as for every ME it provides its type (FORMULA or TERM or part thereof) as well as access functions to retrieve meaningful sub-components (left/right-hand side, (nested) bracketed sub-expressions, etc.) The former is used to "normalise" input for parsing (MEs are substituted by their symbolic types), the latter is used by the anaphor resolution module to resolve references to parts of mathematical notation.

Next, normalised input is **parsed** by a Combinatory Categorial Grammar parser, openccg (`openccg.sourceforge.net`). The parser's resources (the vocabulary and the syntactic categories) are specific to the input's language and the mathematical domain. In particular, the grammar defines a number

of syntactic categories for MEs and their structural parts; for instance, to account for interaction with preceding context in (2). The parser's output is a representation of utterance's linguistic meaning (based on the Praguian tectogrammatical dependency level [7]) which needs to be further interpreted.

The **interpretation** component adds domain specific meaning to the predicates and the relations in the dependency tree, in a step-wise fashion. First, a language-specific semantic lexicon is consulted to assign frame semantics-like roles and concepts (e.g. verb "contain", with symbolic meaning **contain'**, evokes the concept *Containment* with the dependents in the roles *Container* and *Contents*). Next, the concepts are interpreted within the mathematical domain through a linguistically motivated however domain-specific ontology (e.g. *Containment*, in the context of set theory, can be interpreted as a (proper) subset or membership).[3]

The output of the three processing steps is a semantic dependency tree "annotated" with more specific semantic information at the level of 1) concepts (still domain-independent) and 2) domain-specific interpretations (possibly ambiguous). Discourse referents at the tree nodes can be further processed by an anaphor resolution component. The output represents only the *linguistically realised* content; the language processing system is not in a position to reason about the logical validity of the proof steps which the utterances express. However, the tree can be transformed into a representation for further processing, for instance, by an automated theorem prover.

4 Related Work and Outlook

The idea of automating interpretation of mathematical discourse is not new. Baur [1] and Zinn [11] present DRT-based approaches [5] in integrated architectures. Their systems process example sentences (Zinn's system can process two proofs; nine sentences) in English after the text has been manually segmented and mathematical expressions manually marked, and they do not address imprecise language. MARACHNA [3] has a modular architecture similar to ours, however, it does not handle interleaved natural and symbolic language in a systematic way and does not address imprecision either. Kamareddine et al. [4] present a system which requires that the mathematical input text be manually annotated before any processing can take place, however, once the text is annotated, a number of transformations of the original text automate the analysis, upon which the content can be automatically verified. Our approach can be thought of as being complementary to this, in that [4] address informal mathematical expressions in a principled way, whereas their treatment of natural language phenomena is rather ad hoc by comparison with ours.

Our current implementation of the pre-processing, parsing, lexical semantic and interpretation modules is informed by and tested on *syntactically well-formed*

[3] More details on parsing can be found in [8], on lexical semantic analysis in [9] and on interpretation in [2].

utterances from German corpora of dialogues on tutoring proofs,[4] however, we believe that it lends itself also to processing textbook proofs. We are presently extending grammar resources to evaluate its performance on proofs from LaTeX source of Landau's Grundlagen.[5] In the future we are also planning to test the system on English OCR'ed text. Further, we are implementing an anaphor resolution algorithm for mathematical discourse informed by a corpus analysis [10].

References

1. Baur, J. (1999). Syntax und Semantik mathematischer Texte. Diplomarbeit. Computerlinguistik, Universität des Saarlandes, Saarbrücken, Germany.
2. Horacek, H., Wolska, M. (2006). Interpreting semi-formal utterances in dialogs about mathematical proofs. *Data and Knowledge Engineering*, 58(1):90-106.
3. Natho, N., Jeschke, S., Pfeiffer, O. and Wilke, M. (2008) Natural language processing methods for extracting information from mathematical texts *Advances in Communication Systems and Electrical Engineering*, LNEE 4, pp. 297–308.
4. Kamareddine, F., Lamar, R., Maarek, M., Wells, J. B. (2007) Restoring Natural Language as a Computerised Mathematics Input Method LNCS 4573, pp. 280–295.
5. Kamp H., Reyle U. (1993). From Discourse to Logic. Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer, Dordrecht.
6. Lakoff, G., Núñez, R. E. (2000). Where mathematics comes from. How the embodied mind brings mathematics into being. New York: Basic Books.
7. Sgall, P., Hajičová, E., Panevová, J. (1986) The meaning of the sentence in its semantic and pragmatic aspects. Dordrecht: Reidel.
8. Wolska, M., Kruijff-Korbayová (2004). Analysis of mixed natural and symbolic language input in mathematical dialogs. In *Proceedings of ACL-04*, pp. 25–32.
9. Wolska, M., Kruijff-Korbayová, I., and Horacek, H. (2004). Lexical-semantic interpretation of language input in mathematical dialogs. In *Proceedings of the ACL 2nd Workshop on Text Meaning and Interpretation*, pp. 81–88.
10. Wolska, M., Kruijff-Korbayová, I. (2006). Modeling anaphora in informal mathematical dialogue. In *Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue (brandial-06)*, pp. 147–154.
11. Zinn, C. (2006). Supporting the formal verification of mathematical texts Journal of Applied Logic, 4(4), pp. 592–621.

[4] `www.ags.uni-sb.de/~dialog/www/studies.php`

[5] `www.cs.ru.nl/~freek/aut/grundlagen-1.0.tar.gz`