

The Naproche Project

Controlled Natural Language Proof Checking of Mathematical Texts

M. Cramer, P. Koepke, D. Kühlwein, B. Schröder, and J. Veldman

University of Bonn and University of Duisburg-Essen
{cramer, koepke, kuehlwei, veldman}@math.uni-bonn.de,
bernhard.schroeder@uni-due.de
<http://www.naproche.net>

1 Outline of the Naproche Project

The Naproche project¹ (NAtural language PROOf CHEcking) studies the semi-formal language of mathematics (SFLM) as used in journals and textbooks from the perspectives of linguistics, logic and mathematics. A central goal of Naproche is to develop and implement a controlled natural language (CNL) for mathematical texts which can be transformed automatically into equivalent first order formulas by methods of computational linguistics. Since parts of mathematical texts are strictly formal, we expect that a mathematical CNL can cover many aspects of SFLM.

Further work in the project is directed towards automatic manipulation and proof checking of CNL texts and the translation of extensive mathematical theories into the Naproche CNL.

2 The Semi-Formal Language of Mathematics

As an example of the semi-formal language of mathematics (SFLM), we cite a proof for the theorem “ $\sqrt{2}$ is irrational” from HARDY-WRIGHT’s introduction to number theory [2].

If $\sqrt{2}$ is rational, then the equation $a^2 = 2b^2$ is soluble in integers a, b with $(a, b) = 1$. Hence a^2 is even, and therefore a is even. If $a = 2c$, then $4c^2 = 2b^2$, $2c^2 = b^2$, and b is also even, contrary to the hypothesis that $(a, b) = 1$.

SFLM incorporates the syntax and semantics of the general natural language, so that it takes over its complexity and some of its ambiguities. However, SFLM texts are distinguished from common language texts by several characteristics:

¹ Naproche is a joint initiative of PETER KOEPKE (Mathematics, University of Bonn) and BERNHARD SCHRÖDER (Linguistics, University of Duisburg-Essen).

- They combine natural language expressions with mathematical symbols and formulas, which can syntactically function like noun phrases or sub-propositions.
- Constructions which are hard to disambiguate are generally avoided.
- Mathematical symbols can be used for disambiguation, e.g. by use of variables instead of anaphoric pronouns.
- Assumptions can be introduced and retracted. For example, the proof cited above is a proof by contradiction: At the beginning, it is assumed that $\sqrt{2}$ is rational. The claims that follow are understood to be relativised to this assumption. Finally the assumption leads to a contradiction, and is retracted to conclude that $\sqrt{2}$ is irrational.
- Mathematical texts are highly structured. At a global level, they are commonly divided into building blocks like definitions, lemmas, theorems and proofs. Inside a proof, assumptions can be nested into other assumptions, so that the scopes of assumptions define a hierarchical proof structure.
- Definitions add new symbols and expressions to the vocabulary and fix their meaning.
- Proof steps are commonly justified by referring to results in other texts, or previous passages in the same text. So there are intertextual and intratextual references.

3 The Naproche CNL

The Naproche CNL is currently a small subset of SFLM which includes some of the above mentioned characteristics of SFLM.

A Naproche text is structured by structure markers: *Theorem*, *Lemma*, *Proof* and *Qed*. A theorem is presented after the structure marker *Theorem*, and its proof follows between the structure markers *Proof* and *Qed*.

First order formulas can be combined with natural language expressions to form statements, definitions and assumptions.

Assumptions are always opened by an assumption trigger (e.g. *let*, *consider*, *assume that ...*), and closed by a sentence starting with *thus* or by a *Qed*. Definitions always start with *define*. Assertions are made by statements starting with a statement trigger (*then*, *hence*, *therefore*, *so*, ...) followed by a first order formula or a naturally predicated term. Additionally, negation, quantification and implication may be expressed in natural language.

The proof cited above can be reformulated in the Naproche CNL as follows:

Theorem. $\sqrt{2}$ is irrational.

Proof.

Assume that $\sqrt{2}$ is rational. Then there are integers a, b such that $a^2 = 2b^2$ and $(a, b) = 1$. Hence a^2 is even. Therefore a is even. So there is an integer c such that $a = 2c$. Then $4c^2 = 2b^2$ and $2c^2 = b^2$. So b is even. Contradiction. *Qed.*

This is the actual graphical appearance of the text, since Naproche uses the mathematical WYSIWYG editor TeXmacs.

4 Proof Representation Structures

Texts written in the Naproche CNL are translated into *Proof Representation Structures* or PRSs (see [4], [6]). PRSs are Discourse Representation Structures (DRSs, [3]), which are enriched in such a way as to represent the distinguishing characteristics of SFLM discussed in section 2.

A PRS has five constituents: An identification number, a list of discourse referents, a list of mathematical referents, a list of textual referents and an ordered list of conditions. Similar to DRSs, we can display PRSs as “boxes” (Figure 1).

Mathematical referents are the terms and formulas which appear in the text. As in DRSs, discourse referents are used to identify objects in the domain of the discourse. However, the domain contains two kinds of objects: mathematical objects like numbers or sets, and the symbols and formulas which are used to refer to or make claims about mathematical objects. Discourse referents can identify objects of either kind.

PRSs have identification numbers, so that they can be referred to at some later point. The textual referents indicate the intratextual and intertextual references.

Just as in the case of DRSs, PRSs and PRS conditions have to be defined recursively: Let A, B be PRSs, X, X_1, \dots, X_n discourse referents and Y a mathematical referent. Then

- for any n -ary predicate p (e.g. expressed by adjectives and noun phrases in predicative use and verbs in SFLM), $p(X_1, \dots, X_n)$ is a condition.
- $holds(X)$ is a condition representing the claim that the formula referenced by X is true.
- $math_id(X, Y)$ is a condition which binds a discourse referent to a mathematical referent (a formula or a term).
- A is a condition.
- $\neg A$ is a condition, representing a negation.
- $A := B$ is a condition, representing a definition.
- $A \Rightarrow B$ is a condition, representing an assumption (A) and the set of claims made inside the scope of this assumption (B).
- *contradiction* is a condition, representing a contradiction.

Note that contrary to the case of DRSs, a bare PRS can be a direct condition of a PRS. This allows to represent in a PRS the structure of a text divided into building blocks (definitions, lemmas, theorems, proofs) by structure markers. The hierarchical structure of assumptions is represented by nesting conditions of the form $A \Rightarrow B$: A contains an assumption, and B contains the representation of all claims made inside the scope of that assumption.

i	
d_1, \dots, d_n	m_1, \dots, m_k
c_1	
\vdots	
c_l	
r_1, \dots, r_p	

Fig. 1. A PRS with identification number i , discourse referents d_1, \dots, d_n , mathematical referents m_1, \dots, m_k , conditions c_1, \dots, c_l and textual referents r_1, \dots, r_p

The algorithm creating PRSs from CNL proceeds sequentially: It starts with the empty PRS. Each sentence or structure marker in the discourse updates the PRS according to an algorithm similar to a standard DRS construction algorithm but taking the nesting of assumptions into account. [4]

The PRS constructed from the example proof is shown on the right (abbreviated and without the empty reference slots).

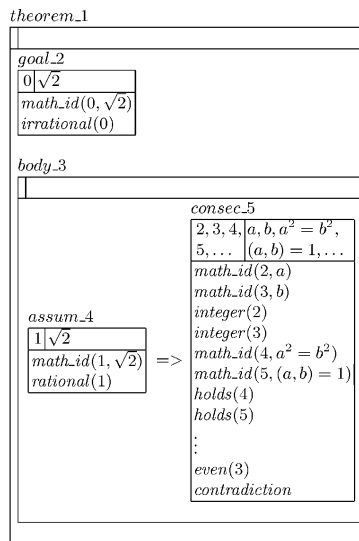
5 Translating PRSs into First Order Logic

PRSs, like DRSs, have a canonical translation into first order logic and can be processed further by logical tools. This translation is performed in a way similar to the DRS to first-order translation described in [1]. For example, the discourse referents normally trigger existential quantifiers, but in a PRS condition of the form $A \Rightarrow B$ the discourse referents of A trigger universal quantifiers. There are two main differences between PRS to first-order translation and DRS to first-order translation that need discussion:

- The *math_id*-conditions are not translated into subformulas of the first-order translation. A *math_id*-condition which binds a discourse referent to a term triggers a substitution of that discourse referent by that term in the translation of subsequent conditions. A *math_id*-condition which binds a discourse referent to a formula causes a subsequent *holds*-condition with that discourse referent to be translated by that formula.
- Definition conditions are also not translated into subformulas of the translation. Instead, a condition of the form $A := B$ triggers a substitution of the relation symbol it defines by the first-order translation of B in subsequent conditions.

6 Results

We created a program, the Naproche system, which can parse an earlier version of the Naproche CNL, create the appropriate PRSs, and check them for correctness using automated theorem provers. The BURALI-FORTI paradox, a well known mathematical theorem, as well as several basic statements in group theory, were formulated in the Naproche CNL, and checked for correctness using the Naproche system.



7 Related and Future Work

CLAUS ZINN wrote his PhD thesis about informal mathematical discourse [8]. He used an extended version of DRSs, which he also called Proof Representation Structures, to represent the meaning of a text. The PRS definitions of Naproche and ZINN are different.

We are collaborating with the VeriMathDoc project [7], which is developing a mathematical assistant system that naturally supports mathematicians in the development, authoring and publication of mathematical work.

We shall extend the Naproche controlled natural language to include a rich mathematical formula language and many argumentative mathematical phrases and constructs. Once the Naproche CNL is sufficiently extensive and powerful, we shall reformulate and check proofs from various areas of mathematics in a way “understandable” to men and machines.

References

1. Blackburn, P., Bos, J.: Representation and Inference for Natural Language Volume II: Working with Discourse Representation Structures (1999)
2. Hardy, G. H., Wright, E. M.: An Introduction to the Theory of Numbers (Fourth Edition, 1960)
3. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Kluwer Academic Publisher (1993)
4. Kolev, N.: Generating Proof Representation Structures for the Project Naproche, Magister thesis (University of Bonn, 2008), <http://naproche.net/downloads.shtml>
5. Kühlwein, D.: A short introduction to Naproche v0.1, <http://naproche.net/downloads.shtml>
6. Kühlwein, D.: A calculus for Proof Representation Structures, Diploma thesis (University of Bonn, 2008), <http://naproche.net/downloads.shtml>
7. VeriMathDoc website: <http://www.ags.uni-sb.de/~afiedler/verimathdoc/>
8. Zinn, C.: Understanding Informal Mathematical Discourse, PhD thesis at the University of Erlangen (2004), <http://citeseer.ist.psu.edu/233023.html>